

國立台灣藝術大學 多媒體動畫藝術系碩士班

96 學年度第 1 學期

【多媒體創作】課程講義

--本課程獲教育部顧問室中綱計畫「人文數位教學」支持--

授課教師 陳永賢 編

國立台灣藝術大學 多媒體動畫藝術系碩士班
96 學年度第 1 學期【多媒體創作】課程講義
--本課程獲教育部顧問室中綱計畫「人文數位教學」支持--

目錄

第一篇 多媒體藝術評介

第一章、互動式媒體藝術創作觀念之探討

第二章、禪的思維與新媒體藝術

第三章、數位奇觀與新互動倫理

第四章、感官與情境交融的互動裝置

第五章、數位時代的動漫快感

第二篇 程式設計與多媒體創作

第一章、Processing 程式概論

第二章、Processing 程式與軟體安裝

第三章、Processing 程式指令與說明

第四章、Java ME 程式與說明

第五章、Mobile Processing 軟體安裝

第六章、Mobile Processing 開發程序

第一篇 多媒體藝術評介

第一章、互動式媒體藝術創作觀念之探討

第二章、禪的思維與新媒體藝術

第三章、數位奇觀與新互動倫理

第四章、感官與情境交融的互動裝置

第五章、數位時代的動漫快感

講義內容由陳永賢教授撰文

第二篇 程式設計與多媒體創作

第一章、Processing 程式概論

第二章、Processing 程式與軟體安裝

第三章、Processing 程式指令與說明

第四章、Java ME 程式與說明

第五章、Mobile Processing 軟體安裝

第六章、Mobile Processing 開發程序

【 Webcam 與多媒體創作 】【 Mobile 與多媒體創作 】 工作坊

課程主持：陳永賢教授／工作坊教師：李家祥老師

（程式講義由李家祥老師提供）

第一章、互動式媒體藝術創作觀念之探討

撰文/陳永賢教授

論文出處：

陳永賢。〈互動式媒體藝術創作觀念之探討〉。《藝術學報》。台北：台灣藝術大學出版，2005年10月，第77期)：51-66。

摘要

因科技時代帶來豐富的媒材，使當代藝術家逐漸結合文本、音樂、繪畫、設計、錄像，製作出跨領域的藝術形式，也呈現出新穎性與獨特的視覺力量。藝術家運用科技工具產生複合性的作品，脫離了原始純藝術創作的範疇，跨領域介入娛樂、遊戲、學習等數位傳播的內容產業。如此，可探討的是，當代藝術的跨媒材發展，何以滲入上述所列的數位傳播內容？從本文的研究中，將從藝術中的遊戲觀念脈絡出發，進而分析新媒體藝術以互動式創作呈現時所產生的創作思考與美學觀點。

關鍵詞：互動式、新媒體藝術、當代藝術

Abstract :

The advent of new technologies in the field of contemporary art opened up the possibility for artists to combine a variety of media, including text, music, painting, design and video or film, to make new, interdisciplinary artforms with new and unique energies. The diversity of forms used to create these multimedia works means that the response to the work is equally diverse - it can be seen as pure fun, or a game, or thought-provoking, or educational or simply aesthetically pleasing. But how are those different media combined to create that multiplicity of responses? This study will analyse the concept of interactivity in contemporary art, bringing together the inspirational, the creative and the aesthetic components of the form, by investigating the concept of new media art.

Keywords: Interactive; New Media Art; Contemporary Art

壹、遊戲、互動與藝術創作之關係

從藝術發展的角度來看，藝術起源從遊戲說由來已久，如康德(Immanuel Kant, 1724-1804)提出「藝術遊戲說」，強調自由是藝術的精髓，把藝術歸結為遊戲；席勒(Friedrich Schiller, 1759-1805)提出藝術是「遊戲衝動說」；斯賓塞(Herbert Spencen, 1820-1903)把藝術看作遊戲，他認為「遊戲是精力過剩」等論述，藝術創作被認為起源於遊戲，大都從主觀的人文學或藝術哲學所探究出來的，即是一種內在哲理性的陳述。而近代對於這種將內隱的動機論之外，實質展現在外顯的藝術創作上，則要推至達達藝術家杜象(Marcel Duchamp)的現成物觀念。

杜象與達達主義的藝術家們進行不同方式的批評、挑釁、玩笑，在現實生活中摻雜一些虛無主義的觀念來反抗社會衝突，同時也顛覆了傳統窠臼裡的價值觀。一九一九年，杜象在達文西名作〈蒙娜麗莎的微笑〉畫上鬍鬚，以〈L.H.O.O.Q.〉大膽揭示藝術的遊戲精神外，卸除藝術原有的嚴肅性，並劃破觀眾對於「審美」的定義。因而，觀眾和作品之間的關係也開始產生微妙變化。杜象認為，創作行為並不是由藝術家個人獨自呈現，必須加上觀眾的參與和詮釋，使作品與外在世界產生關連並造就其意義。同樣的，達達主義「開藝術一個玩笑」的作風，釋放出另一種以調侃或幽默為基調的因子，使藝術品與觀眾的距離越來越接近，即時解除那道被認為難以理解的神秘鴻溝。

若以藝術的發展脈絡來看，二〇、三〇年代的藝術遊戲觀念，特別是呈現物件的遊戲性格方面，它不僅承襲了物體藝術的意圖，而藝術家創造出大型的玩具式物件用以滿足樂喜分享，同時也製造出屬於玩物樂趣的追求遠勝於調侃式諷語。例如，奧斯卡·雪勒蒙(Oskar Schlemmer)的作品〈潛水人〉(The Diver, 1922)(圖1)，直接組裝人偶造型，加上環繞式的帶狀物裝飾，給予形塑的大玩具繽紛色彩，觀賞者莫不留下愉悅的心情。

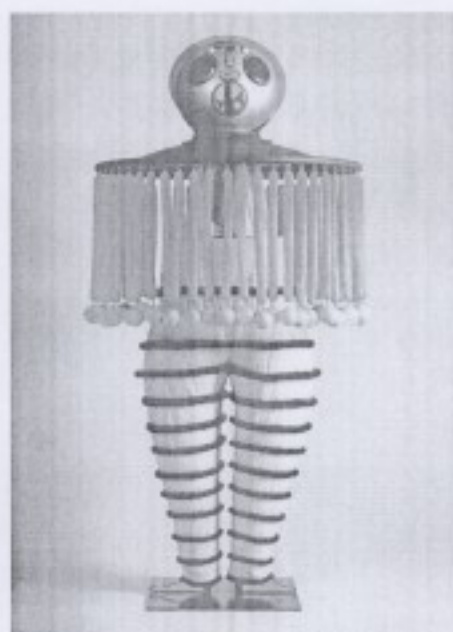


圖1 奧斯卡·雪勒蒙(Oskar Schlemmer)的作品〈潛水人〉(The Diver, 1922)

四〇、五〇年代，繼承遊戲觀念的藝術家再度推展，從現成物的喜好延伸到物體藝術的嘗試，尤其是法國新寫實主義（*Nouveau Realisme*）對日常事物的觀察進而以此為創作主題，他們反對抽象藝術之想像而內省的自我滿足。例如丁格利（*J. Tinguely*）集結一些拾獲而得的廢棄物品，拼湊而成如 *Willy Rotzler*（1996）所敘述的「極具戲劇性，姿勢又帶悲劇或怪異性，而令人感染其歡樂氣氛。」這種打破日常生活陳規的思考方式，在某方面來說具有物件迷戀與觀眾參與的特質，也解除藝術與生活上的界線，亦如布洛克（*Bazon Brock*）所指的：過去五十年的藝術歷史，完全靠生產者的改變在決定。感受在藝術生產上也扮演重要角色，在偶發事件中，觀眾成為事件進行中必要的參與者，而動態藝術品所要傳達的美感，更需要觀眾在作品前走動。他說：「從我的作品來看，我可以確定未來五十年藝術發展的變化，將由觀者、參與者、感受等方面出發。美學實務也將從傳達行為形式，成為駕馭社會的實務。」

接著，於六〇、七〇年代「開藝術玩笑」的精神不斷擴張，經過同一時期錄像藝術、觀念藝術和行為藝術的洗禮，藝術家對藝術創作的遊戲態度，添加了一種冒險和實驗精神，進而延伸至「為遊戲而遊戲」的創作試探上。例如丹·葛瑞漢（*Dan Graham*）的〈現在進行中的過去〉（*Present Continuous Past*, 1974）（圖2），便是結合多媒材的使用，讓觀眾進入一個由鏡面所搭建的密閉空間，經由監視器和螢幕延遲播放的效果，使觀賞者看到自己在十秒以前的動作，於是開始有人不斷跳躍或是扮鬼臉，再來欣賞這段延遲畫面，當然，空間裡也不斷傳出逗趣、互動、好玩的笑聲。又如丹尼斯·歐片罕（*Dennis Oppenheim*）〈大音階敲擊的主題〉（*Theme for a Major Hit*, 1974）（圖3），利用機動原理，將懸絲的特製玩偶搭配在一起，隨著旁邊的音樂旋律，三個身著西裝的人形偶像不停地躁動步伐，像是一場正在進行中的舞會，節拍、音樂、律動、歡樂的情境，讓人想要加入他們婆娑起舞的行列。



圖2 丹·葛瑞漢（*Dan Graham*）的作品〈現在進行中的過去〉（*Present Continuous Past*, 1974）



圖3 丹尼斯·歐片罕 (Dennis Oppenheim) 的作品〈大音階敲擊的主題〉(Theme for a Major Hit, 1974)

八〇年代以後，由於科技媒體的普及與便利性而延生出各式各樣的電子、網路、數位藝術等新潮流，其遊戲風格蔚為時下的流行語法，因此遊戲性格的概念不僅擺脫制式的陳規，也因而成為藝術家創作的主题之一。

貳、互動性媒體藝術的當代脈絡

就整個視覺藝術的發展來說，遊戲的內容並無法就其形式來作一個明顯的歸類，它是一種趣味感的追求，或者創作者以好玩、談諧、幽默的創作心情詮釋，所呈現的效果也因觀眾或參與者個人的感受程度而有不同的差異。因此，互動式的作品呈現，在媒材使用、表達觀念與觀眾溝通的時間過程中，成為當代藝術家熱衷追求的創作形式之一。

今日，由於電腦技術、數位化圖像處理、虛擬實境的技術日趨成熟，進而帶動媒體藝術多元面向的發展，而從藝術被科技化與科技被藝術化，這種混雜且相互牽涉的媒體藝術新象中，傳達出的哲學概念，可從班雅明 (Walter Benjamin) 的論述為基礎，他在一九三六年即提出以「電影媒體」作為時代新藝術的例證，認為包含聲音傳達與移動影像的電影視覺和一般傳統平面的繪畫相比較，更能精確地陳述自我生存環境的描述和感官訊息的傳遞。因此，當班雅明「機械複製」的觀點成為藝術批判的辯證，科學介入創作，早以深刻地進入並影響到當代藝術創作的領域。

此外，特別以互動式媒體藝術作品的發展來看，藝術家結合科技跨領域呈現多元化的創作

現象，從一九八〇年代末傑佛瑞·蕭(Jeffery Shaw)的《可讀的城市》(The Legible City, 1988-91) (圖4)讓觀眾坐上腳踏車，透過虛擬的三度空間、電腦及時運算等技術，使觀者介入並產生幻覺場景，明確地建立以互動為主體的美學形式。九〇年代的藝術家，利用更成熟的數位運算技術，開放觀眾進入一個特殊的環境之中，參與一場更為細緻的互動時空，例如松墨爾與米諾奴(Christa Sommerer & Laurent Mignonneau)的作品《互動植物生長計畫》(Interactive Plant Growing, 1993) (圖5)，藝術家在空間中裝置了五盆真實的綠色植物，透過感應器與電腦系統的連結運算程式，當觀者觸摸這些植物時，同時間在前方的大型螢幕上，亦投影這些植物「生長」的虛擬影像，不同參與者所觸摸的植物，將有不同的植物影像不斷的堆疊上去，形成綠色叢林中有著真實與虛擬相互對應的植物成長紀錄。

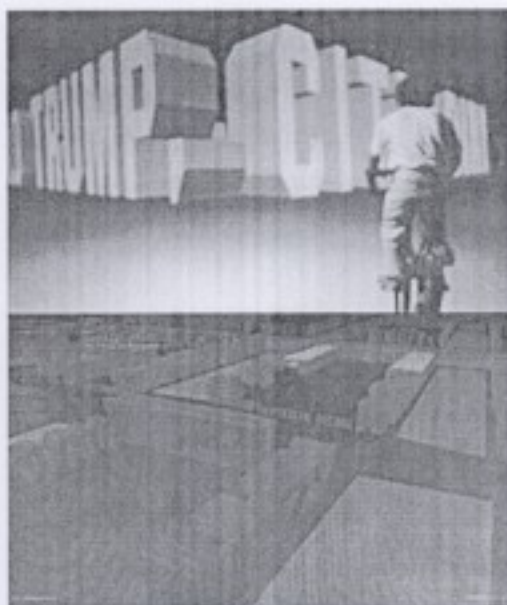


圖4 傑佛瑞·蕭(Jeffery Shaw)的作品《可讀的城市》(The Legible City, 1988-91)

資料來源：Michael Rush, (1999), "New Media in Late 20th-Century Art." London: Thames & Hudson.

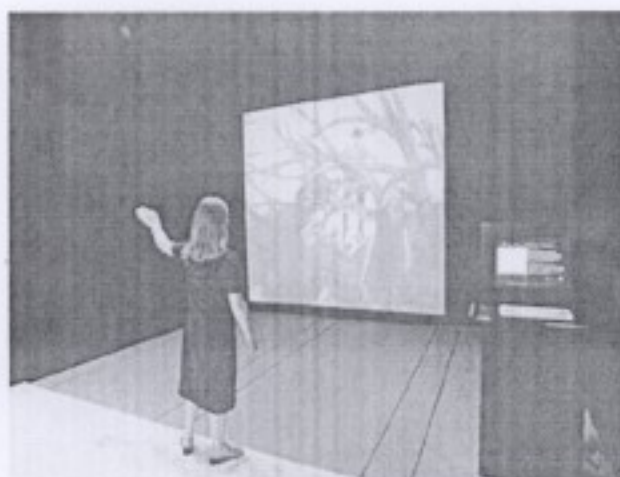


圖5 松墨爾與米諾奴(Christa Sommerer & Laurent Mignonneau)的作品《互動植物生長計畫》(Interactive Plant Growing, 1993)

資料來源：http://www.aec.at/de/archives/picture_ausgabe_02_new.asp?iArealID=273&showArealID=273

再如，日本藝術家岩井俊雄（Toshio Iwai）和藤幡正樹（Massaki Fujihata）的創作亦需要觀眾的介入來完成作品。以岩井俊雄的〈鋼琴：作為影像的媒體〉（Piano—as Image Media, 1995）（圖 6）來說，觀眾透過滑鼠控制音符的對應關係，電腦記憶滑鼠軌跡而產生光點，依此光點符號擲向琴鍵並發出鋼琴的旋律，讓參與者成為一位及時性的作曲家，填寫了可以立刻演奏的樂章後，滿足音樂躍動的快感。再者，藤幡正樹的互動裝置〈冊頁之外〉（Beyond Pages, 1995）（圖 7），作品現場是一個像書房般的幽暗空間，空間中置放著實際物體的書桌椅和檯燈，桌面上投影著一本虛擬的精裝書籍，觀眾可以拿起桌上的數位筆翻開封面，緊接著出現一頁頁文字與物體並列的內容，當觀眾以筆觸碰書中所呈現物體時，它即產生互動反應，如蘋果被一口口的咬掉、追逐逃出書外的石頭、文字字母發音，或是參與書中的電燈開關來點亮桌上的檯燈，以及打開門把讓小孩進入室內空間。觀眾無不在驚奇與歡樂的氣氛下，與這本神奇之書進行愉悅的對談互動。

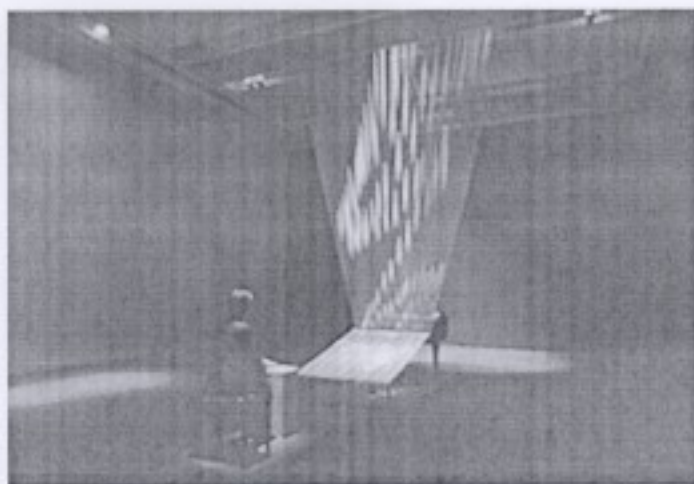


圖 6 岩井俊雄（Toshio Iwai）的作品〈鋼琴：作為影像的媒體〉（Piano—as Image Media, 1995）
資料來源：<<http://addendum.mit.edu/e-journals/Leonardo/gallery/gallery343/iwai2.html>>

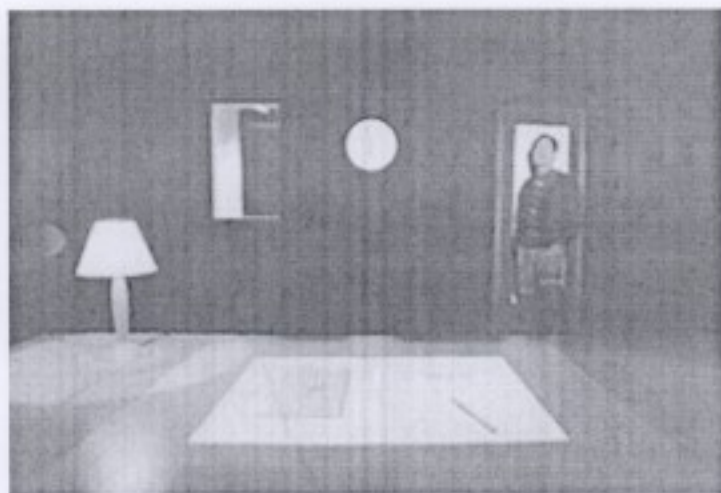


圖 7 藤幡正樹（Massaki Fujihata）的作品〈冊頁之外〉（Beyond Pages, 1995）
資料來源：<http://www.iamas.ac.jp/interaction/i97/artist_Fujihata.html>

此外，墨西哥裔加拿大籍藝術家拉法耶·羅札諾－漢墨 (Rafael Lozano-Hemmer)，他的作品如〈維多利亞高地，對應建築 4〉(2000) 是為千禧年的來臨而作，讓全世界觀眾可透過網際網路的傳輸功能，去遙控墨西哥市卓卡洛廣場上的十八座探照燈，使其照明有不同的轉向。〈身體電影，對應建築 6〉(2001) (圖 8)，是為歐洲鹿特丹城市所作，其應用手法是把來往的民衆身影投射到二十五公尺高的建築外牆上，現場觀眾可利用自己的身體影子與投影人物互動遊戲。〈頻率與聲音，對應建築 9〉則是以觀眾的影子和聲音的互動為概念，當觀眾的影子經過強光照射而投射在牆面上並產生了座標，以牆上的座標為依據的收音機，會隨著觀者的腳步移動而不斷地切換電台頻道，影子的大小則控制電台頻道所發出音量。此系統由空間、光影、電頻、電訊控制系統所組成，在機器的嚴密控制下，影子與聲音成為觀眾遊戲的來源，並造成空間與人物之間互動的樂趣。



圖 8 拉法耶·羅札諾－漢墨 (Rafael Lozano-Hemmer) 的作品〈身體電影，對應建築 6〉

資料來源：<<http://www.fundacion.telefonica.com/at/rh/index.html>>

互動式媒體藝術的創作觀念，無疑是藝術家掌握作品呈現時重要的關鍵，其「互動性」(Interactivity) 遂成為新媒體藝術中，最具精髓的名詞，亦是一種權力與控制的新語彙。對於創作互動式的必要性，Itsuo Sakane 認為：「互動藝術的行為模式呈現多元的藝術型態，觀眾和藝術家在互動的過程中使作品產生新的意義。」其實，藝術作品「互動性」不僅是功能上的需求，也是「德性」的要求，譬如從服務人員到錄音播放之制式化的禮貌用語，人性互動至此蛻變成系統性的互動，科技藝術的發展也無疑面對這種局勢，並有所回應。

然而，互動式媒體藝術雖然是當代藝術正在進行的一種狀態，也是一種時代趨勢。但如 Louise Dompierre (1995) 所指的，「過渡的互動形式作品讓觀眾迷失於藝術本質，甚至忽略其創作意涵。」而藝術家的角色打破以往唯一詮釋者的特質，當觀眾參與介入，產生的自我藝術解讀，是否產生的藝術詮釋的困難。如此，我們不禁要思考，藝術創作與互動環境之間，誰是創造者或詮釋者角色？以及如何以互動式創作擴充其藝術本質的內涵？

參、互動性之於媒體藝術的創作思維

如上所述，藝術家藉由各種媒介進行互動裝置、電腦多媒體和網路連結等多元形式的創作，其考慮的前提不外乎是誰來參與？誰來互動？為何互動？假如在預設的互動範圍內，是否能夠達到預期的目標，這些都是創作者所關心的問題。

因此，Krzysztof Wodiczko（1983）在其〈Public Projection〉的論述中談到：「媒體藝術並不只是在於一種場域的力量，更重要的是在其媒體屬性的延展性，如此便可以提供介於個別慾望和模擬象徵的力量。」他的想法給予創作者一個更宏觀的思考，也對互動性的內涵提出新的見解。相對的，對於互動性本質與創作意念彼此間相互結合時，藝術創作上如何思考此兩者的緊密關係？

一、互動性本質與創作之關連

當藝術家創作的作品落實於互動式的模式時，藝術品的本質已經跳脫原本被動的觀賞角色，進而成為另一種審美經驗的特質。因此，互動式媒體藝術的創作思考也已進入觀眾反應、介入詮釋的角色互換，其視覺傳達的方式亦隨之改變。

G. Novakovic（2003）透過互動式的創作語言指出，電腦科技促使互動藝術的發展，互動藝術的隱喻是一種強而有力的儀器，其參與之過程經驗可以協助觀眾免除科技所帶來的恐懼與壓迫，甚至帶領觀眾瞭解日常生活中的資訊運算功能，擴展如何觀賞、如何製作等流程，進而釋放互動藝術中的精神層面。

今日隨著科技時代的進步，電信溝通、互動多媒體、虛擬實境等技術日趨純熟，互動性已經成為日常生活中不可或缺的名詞。然而，當代的藝術家也藉由敏銳觀察，瞭解科技與互動藝術之間的彼此消長與衝擊，進而改變互動性的實際意涵。對於互動概念與詮釋方法，可作三種解釋：

- （一）、參與者與參與者之間的互動關係：藉由參與者（觀眾）之間的溝通與共同經歷的時間過程，兩者都可獲得彼此回饋的體驗。
- （二）、參與者與物體之間的關係：經由電腦資訊的介面指示，參與者得到指引訊息之後，能夠和此一物體進行某種程度上的心得交換，進而達到參與後的經驗與回饋。這裡所談的訊息（Information），指的是針對參與者而發出的，在交換訊息時，此兩者可以按照彼此反應而調整其傳遞訊息的方式。此外，溝通的管道必須是雙向的，便於保持其順暢性。
- （三）、物體與物體之間的關連意涵：藉由兩種或兩種物體（如電腦、儀器、經過設定的程式或特定器材）產生相互影響，其中這些物體彼此有著供需狀態的行為模式。

二、互動式媒體藝術的構成要件

互動式藝術的表達，狹義來說是包含觀眾參與其中的藝術呈現，而廣義的定義則是將觀看以及瞭解作品的過程也視為是一種參與的互動行為。因此，Robert Jacobson (1999) 提出六種元素作為互動的構成基礎，包括：回饋、控制、創造、生產力、溝通、適應性等，以視為作品是否合乎互動性的標準。分述如下：

- (一)、回饋 (Feedback)：互動的過程產生了一種回饋的表徵，雖然使用者存在著個別差異，但經由個人的回應或動作表現，卻可滿足於個別化、適性化的回饋條件。
- (二)、控制 (Control)：在於接收訊息或程式路徑的規劃與設計上，除了一般的系統調整外，非線性資訊處理系統亦不可或缺，它是整體操控的中樞，也是系統運作的核心。
- (三)、創造 (Creativity)：觀眾於參與的過程中，藉由其系統介面或影像條件得到另一種思考空間，使原本屬於被動的觀賞者成為積極參與的創造者，享受創造性的樂趣。
- (四)、生產力 (Productivity)：系統介面提供觀眾成為參與行為之後，兩者之間達成互動的共識，並傾向達到某種生產形式的運作目標。
- (五)、溝通 (Communications)：建立雙向溝通的傳達管道，使提供各種訊息時，觀者能夠表達其需求與看法，建立兩者互動的關係與品質。
- (六)、適應性 (Adaptivity)：針對相異的使用者而做出適當調整，在觀者使用習慣或控制介面上，可以啟整不同的資訊及互動方式，讓機器或電腦程式適應於觀者，並製造

除此之外，對於互動介面與視覺影像相互搭配的原則，筆者認為有幾點元素必須納入其中，包括：

- (一)、選擇性 (Options)：當互動式藝術被賦予與觀眾溝通的橋樑時，藝術家藉由各種媒介來處理角色互換，誰是作者的觀念已經被消解，取而代之的是參與者的選擇態度與參與方式。因此，觀眾如何在自由意願下選擇進入並參與其中的互動關係，成為此兩種角色的重要關鍵。
- (二)、隨機性 (Random)：互動性的層面可以是行為動作的表面，如觸發按鈕或點選連結等技術，也可以是心理層面的互動，如回憶、假設等情境。當觀眾介入其中的鏈結區域，必須考量其個體的因素，製造隨機的互動概念以符合使用者的需求。
- (三)、場域性 (Site)：回歸到藝術呈現本身，藝術家在標榜創作動機與互動效果外，亦需製造參與者與作品互動之間的氛圍。無論是室內或戶外的裝置搭配，在此特殊氛圍的場域概念下，更能吸引觀眾融入情境而給予適當的回饋。
- (四)、消費性 (Consumption)：互動式的藝術表現有如祭典儀式的平行概念，經由觀眾參與譯碼之後，容易成為消費性的表徵。因此，如何在消費的心態上填補適當資訊，使觀賞者在互動過程之後有了感官的刺激外，亦能喚起觀者產生內在對話並思考其議題，並回溯於藝術的內涵。

互動式媒體藝術是時代的產物，同時它也破除誰是參與者的角色迷思，如 Lev Manovich

(2000)指出：「人們樂於宣揚新媒體的互動性，但其互動性本來就屬於電腦媒體操控中的一部份，因此，互動性的概念是廣泛而難以界定的。」若以電腦媒體為基礎來討論互動媒體，很容易將焦點置於一種實體的互動概念，因此，盲目地致力於技術的追求將遭時代淘汰，唯有在多元創作的媒材運用上，瞭解互動性的本質與目的，加上人文思維、內涵與深度才能精確地傳達藝術的精髓。

肆、互動式藝術創作的幾種觀照

無可諱言的，當代藝術家藉助新發明的科技器材來創作，其美學觀早已脫離了現代主義之前以藝術家為主的單純自我創作，取而代之的是融合社會意識與觀眾感知的相互互動經驗。今日觀眾與藝術作品間的關係與以往最大不同之處，便在於藝術作品須透過觀眾的參與經驗，才得以延續或累積作品意義，這也是互動式媒體藝術的最大特質。就創作者而言，除了釐清創作觀念、表現目的與執行計畫外，尚必須評估觀眾參與的意願，以及參與動作的後續反應，在整體的考量下，經過縝密的測試與製作，才得以完成互動式的創作過程。因此，互動式的創作內涵包含以下幾項美學特質：

一、遊樂園概念的藝術觀

藝術的遊玩與互動概念，除了凸顯在內在精神與呈現邏輯的現象外，遊戲的性質也代表著所處的時代反思，在社會極度繁榮下，使本來隱藏在文化中的危機和矛盾充分地顯現出來。(陳永賢，2004)

當代學者布希亞 (Jean Baudrillard) 以迪士尼樂園為例，提出「擬像的旋動」(precession des simulacrs) 論述，他認為遊樂園是典型擬像系統的真實表演。在迪士尼樂園裡，以米老鼠為首的卡通家族以及米奇卡通城、美國大街、明日世界、邊疆世界、夢幻王國、動物天地、冒險世界等，觀眾選擇進入遊園時，即宣告進入一場真實世界中的擬像狀態。

學者高宣揚 (2002) 也引述布希亞的論點來談論「消費文化的擬像遊戲性質」，他認為：「當代的擬像遊戲，已經不是某個地方和描述它的地圖之間的關係，不再是原本和副本的關係，不再是實物和境中物之間的關係，也不再是對象和它的概念之間的關係。」因而，遊樂園是典型擬像系統的真實表演，為何彼此產生距離而難以分辨？主要原因可以歸納為：

- (一)、同時存在真實、幻想與虛構，也就是現實生活中涵蓋著未來世界的。換句話說，迪士尼樂園的遊樂世界，是一個以想像為中心的「真實的虛構」，既不是真的，也不是假的，歸根就底說，它是用來征服或鎮攝人心的大機器，它巧妙地裝扮成「兒童式的理想王國」，骨子裡確有一種掩蓋事實的意識型態，這種擬象式的運作，造成虛假和真實之間的混亂，使虛假無法成為虛假，使真實無法成為真實；當然也就是因為虛假的部分無法

被辨識，真實也無法在虛假的狀態中被區分開來而永遠無法真實的呈現。

(二)、生活方式與文化模式的崇拜力量，呈現一個純粹可以想像出來的世界。這個力量在遊樂玩耍時，狂熱的人潮與遊戲場域形成鮮明對比，各種光鮮亮麗的假道具、表演和人群互動激盪起一波波對物質娛樂的好奇，如魔術式的運作，凸顯了真實世界與虛擬世界兩者間價值體系的交互滲透。擬像說的遊樂園觀念和藝術的遊戲、互動概念，有著異曲同工之妙，因為生活和創作彼此交融的過程，導致藝術家擁有經驗性、超驗性的超越慾望，藉由嬉遊的命題來展示批判、顛覆、嘲諷、慾望等元素，進而表現在創作上。

總之，藝術家構築的「遊戲場域」是私密底下的嬉遊心情，雖不及於迪士尼樂園龐大的遊樂核心；然而，解除嚴肅以對的心態，藝術家與創作之間，在玩耍、玩笑、玩物、歡樂之餘，在互動式的觀念不免也透過作品與觀眾間的親密關係達到彼此的供需層次。

二、跨領域概念的藝術觀

媒體藝術源自於多種藝術風格，包括觀念藝術、未來主義宣言、達達式語言、偶發藝術、觀念藝術、行為藝術、錄像藝術、動力藝術、電子藝術、裝置藝術等等，許多藝術家對於讓觀眾參與到作品中深感興趣，而藝術作品本身的定義也不再決定於它的實體與形式，而更多在於它的形成過程和概念陳述。因而，Michael Rush (1999) 將新媒體藝術的內容區分為媒體和表演 (Media And Performance)、錄像藝術 (Video Art)、錄像裝置藝術 (Video Installation Art)、數位藝術 (Digital Art) 等要項，他從人的肢體語言、行為表演在藝術範疇下所形成的特殊符號與新媒體關係作為首要闡述，如早期美國黑山學院由凱吉 (John Cage)、卡彌漢 (Merce Cunningham)、羅斯強伯格 (Robert Rauschenberg) 等三位藝術家結合音樂、舞蹈、視覺藝術的身體互動，一直到布魯斯·紐曼 (Bruce Nauman) 融合機械式器材的影像表演 (Anthro/ Socio, 1992)，進而推展到錄像與錄像裝置、數位藝術、虛擬實境等跨越媒材與技術的表現方式。

其實，這樣的創作已經涵蓋藝術家主觀意念和客觀上的影像、文字、繪畫、雕塑、劇場、表演、建築、電影、聲音、音樂、造形、空間、時間等綜合形式的媒材參與。再者，新世代藝術家對於新科學的隱喻與模式的著迷，尤其是世紀初的量子物理和世紀末的神經科學與生物學，都不斷激發了藝術家的想像力。如此多元的物件屬性和人為因素，其背後也隱含著幾項創作特質：

(一)、跨媒材與跨領域的整合概念

媒體藝術時代的藝術家不再從畫架、畫桌和調色盤上得到滿足，其創作流程也因電腦軟、硬體的操作或互動程式設計、動力操控使用等類別而有不同媒材和領域的區分。如 Julian H. Scaff 延續班雅明 (機械複製時代的藝術作品) 之說法，認為數位複製時代涉及更多層面的議題與創作語彙。因此，媒體藝術逐漸成為跨媒材、跨領域的藝術創作類型，它脫離既定的陳規限制，在作品內容中大膽使用不同的媒材及結合其之跨

領域藝術，採取藝術學之外的知識，如科學、技術、生物學、人類學、社會學、心裡學、政治學……等不同學科，在創作文本中呈現複合性意義而以開展方式反映出一種多元的藝術特質。

(二)、合作結盟的意識型態

由於運用複雜科技工具在藝術創作上有其操作的獨特性與技術性，因此，在跨媒材、跨領域的組合方式上，便有不同工作者以各司其職的角色參與，藉由小組或團體的名義共同創作。此時的「作者權」(Authorship)未必僅有單一作者，因不同領域的藝術工作者共同參與完成，便有多作者(Poly-authorship)的定義。例如一九九三年，英國的一個藝術家小組創立了名為「Technosphere」網站，一個提供線上互動功能的系統，用戶根據系統提供的不同形狀的生物構件以互聯網終端的功能來組建自己的生物，經由傳送到網站來觀察這些由互聯網用戶製造的各種形態生物的相互影響與進化。麻省理工(MIT)的研究人員組成了一個名為「近乎生命」的團體，一九九八年，該團體推出了〈虛擬魚缸系統〉，在人工智慧的系統中，由觀眾觸摸顯示器而生成魚的形狀與成長，展現生命空間的奧妙。在此意識型態之下，跨媒材、跨領域的創作的確需要各類型專業人才，如生物或科學實驗者、程式設計者、音樂家、視覺藝術家等，這種合作結盟因創作所需的關係也更加密切，其藝術之美學樣態即是一種交互主體性(Inter Subjectivity)的表現。

三、情境傳播的藝術觀

當互動式媒體藝術開放觀眾進入參與時，不僅藝術家角色遭受轉變，其作品呈現與參與者行為之間的情境也跟著轉移，譬如說，觀眾進入阿特貝克與阿契托夫(Camille Utterback & Romy Achituv)的作品〈言雨〉(Text Rain)(圖9)，經由隱藏式攝影機擷取參觀者身影而投射在牆上，螢幕畫面顯示漂流而下的字母字串即停留在觀眾的影子上，促使觀眾以不同的肢體動作展開互動對話。此時的情境轉化由觀眾進場開始(進入)，產生好奇(看到自己的影子)、觀察(瞭解自己的影子為何出現在螢幕上)、互動(以肢體動作和文字遊戲)、感受(進一步體驗或享受於自己與作品間的樂趣)、離開作品(離場)等參與過程，幾乎充滿著不同情緒的反應，除了架構參與者形塑作品的平台，其無形而處於內在層次的情境傳播也提供一個重要的氛圍。

另外，從語言分析的角度而言，Lev Manovich 提出建立新媒體藝術的五大原則，包括：數位化呈現(Numeric Representation)、模組性格(Modularity)、自動化操作(Automation)、可變易性(Variability)、轉碼(Transcoding)等，因此，互動式媒體藝術最鮮明的特徵即為其互動性和連結性的應用。藉由互動性和連結性的創作思，考消彌了創作者和觀者所處的情境介面，透過有效的工具、媒材而延伸情境傳播的意義，其情境傳播形式由「作品—連結—互動」、「工具—媒介—轉碼」到「語言—介面—文化」之間，形成相互關連的網絡，同時也將的硬體呈現上的介面，進一步轉換為社會文化意義的觀點上。

以互動式媒體藝術的連結特性來說，呈現的遊戲與創意趣味產生在觀者和作品的互動關係中，觀者並非是旁觀者，而是一個具有操控與自主性格的角色。當觀眾將自身融入作品情境中，與作品系統產生互動，這將導致作品以及個人的意識產生轉化，最後會出現全新的影像、關係、思維與經驗。經由連結、融入、互動、轉化、呈現等不同階段的延伸，觀眾的操控權力頓時大增，他可以使用控制介面，如鍵盤、滑鼠、燈光、聲音、光影感應、抑或其他更複雜精密，甚至是看不見的板機、程式面版、操控感應器等等，進而改變作品的影像、造型、甚至意義。換言之，情境傳播是創作者製造給予的管道，而其情境轉化則交由觀者來承擔，也就是說，觀者和作品間的連結體系與作品引發的轉化情境，是藝術工作者必須考量的問題。

伍、結論

當代的互動式媒體藝術創作，結合以跨媒材、跨領域為整合概念的關係出現，不僅取代傳統上以個人自我為中心的創作範疇，也凸顯出藝術文本的轉變。因而，創造性的藝術實踐並非只從單純的遊戲表象來談論互動的形式，在人文學思考所提供的意涵下，其「單一性」的創作實踐相對於「多元性」、「混合性」的遊戲性格，須以更開放的胸襟接納，藝術創作想像力和實踐力才能相互輝映。

八〇年代以後，由於科技媒體的普及與便利性而延生出各式各樣的電子、網路、數位藝術等新潮流，其遊戲風格蔚為時下的流行語法，因此遊戲性格的概念不僅擺脫制式的陳規，也因而成為藝術家創作的主题之一。

互動式媒體藝術的創作觀念，無疑是藝術家掌握作品呈現時重要的關鍵，其「互動性」(Interactivity) 遂成為新媒體藝術中，最具精髓的名詞，亦是一種權力與控制的新語彙。對於創作互動式的必要性，Itsuo Sakane 認為：「互動藝術的行為模式呈現多元的藝術型態，觀眾和藝術家在互動的過程中使作品產生新的意義。」其實，藝術作品「互動性」不僅是功能上的需求，也是「德性」的要求，譬如從服務人員到錄音播放之制式化的禮貌用語，人性互動至此蛻變成系統性的互動，科技藝術的發展也無疑面對這種局勢，並有所回應。

然而，互動式媒體藝術雖然是當代藝術正在進行的一種狀態，也是一種時代趨勢。但如 Louise Dompierre (1995) 所指的，「過渡的互動形式作品讓觀眾迷失於藝術本質，甚至忽略其創作意涵。」而藝術家的角色打破以往唯一詮釋者的特質，當觀眾參與介入，產生的自我藝術解讀，是否產生的藝術詮釋的困難。如此，我們不禁要思考，藝術創作與互動環境之間，誰是創造者或詮釋者角色？以及如何以互動式創作擴充其藝術本質的內涵？

參考文獻

1. 王俊傑主編 (2004):《漫遊者：國際數位藝術大展》。台中：台灣美術館，。
2. 台北當代藝術館 (2001):《歡樂迷宮》。台北：台北當代藝術館。
3. 林志明 (2004):〈新媒體藝術之「新」〉，《國巨科技藝術國際學術研討會論文集》。台北：國巨文教基金會。
4. 李醒塵 (1996):《西方美學史教程》。台北：淑馨出版社。
5. 吳瑪 譯 (1996):《物體藝術》。台北：遠流出版社。
6. 高宣揚 (2002):《流行文化社會學》。台北：揚智出版社。
7. 陳泰松 (2004):〈在獵戶座之彼端：一種科幻藝術的想像〉，《藝術家》，2004年9月號。
8. 陳永賢 (2004):〈藝術放輕鬆—遊戲場域〉，《藝術家》，頁428~435，2004年4月號。
9. 戴麗卿 (2004):〈數位媒體科技 VS 藝術意識型態〉，《漫遊者：數位媒體的行進與未來—國際論壇峰會論文》。台中：台灣美術館。
10. Benjamin, Walter. (1969). "*The Work of Art in the Age of Mechanical Reproduction.*" New York: Schocken.
11. Dompierre, Louise. (1995). "*Press Enter.*" Canada: Contemporary Art Gallery at Harbourfront Centre.
12. Itsuo Sakane. (1995). "*International Academy of Media Arts and Science.*" Ogaki: IAMAS.
13. Krueger, K.W. (1991). "*Artificial Reality II.*" Addison Wesley Press.
14. Novakovic, G. (2003). "*Electronic Cruelty.*" London: Tate Modern.
15. Poster, Mark. ed.,. (1998). "*Jean Baudrillard: Selected Writings.*" Cambridge: Polity Press.
16. Rush, Michael. (1999). "*New Media in Late 20th-Century Art.*" London: Thames & Hudson.
17. Scaff, Julian H. 'Art and Authenticity in the Age of Digital Reproduction', <<http://www.digitalartsinstitute.com/scaff/>> [1 November 2004].
18. Wodiczko, Krzysztof. (1983). 'Public Project', "Art in Theory 1900-1990." Oxford: Harrison & Wood.
19. ZKM,. (1997). "*Art Practice at the ZKM Institute for Visual Media 1992-1997.*" Ostfildern: Cantz Verlag.

第二章、禪的思維與新媒體藝術—以台灣當代藝術為例

撰文/陳永賢教授

論文出處：

陳永賢。〈禪的思維與新媒體藝術—以台灣當代藝術為例〉。《藝術學報》。台北：台灣藝術大學出版，2007年10月，第81期）：53-69。

摘要

1960年代以來，藉由類比、電子媒介與藝術結合的創作形式逐漸展現新貌；1980年代之後，數位科技、新媒體藝術相繼出現於國際藝術展覽而引起一陣熱潮，這是脫離傳統媒介所引發的效應，一則以新穎的數位素材為號召，其二則包容了令人無法抵抗的臨場感與互動性效果。如今正在蓬勃發展中的新媒體藝術仍然以綜合性的跨領域合作（如藝術家與工程師之間的創作）、新技術的實驗風格（如嶄新的感應裝置與器材設備）、藝術與商業的結合開發（如藝術魅力與流行時尚）、表演形式（如舞台空間與新媒體影像）等，均在創作思維和技術潛能之間形成新的創作類型。新媒體藝術的重要特色之一，即是從觀者參與過程中所轉化的體驗，如觸摸、空間移動、聲音控制、光影感應等互動中獲得即時回饋。然而，正當全球各地不斷發展具有在地精神的電子媒體藝術之時，結合禪思維與新媒體藝術所提供觀者的參悟迴圈，成為另一種可能。因此，本文從兩者之歷史回溯為開端，並藉由台灣藝術家作品分析與禪思意涵等層面，探測其相互關係與未來性。

關鍵詞：禪的思維、新媒體藝術、互動裝置、當代藝術

Abstract

Ever since the information revolution after 1960s, creation via digital computer design has been a significant form in contemporary art. However, the controversies surrounding this new form of art have never ceased after video and new media art made their fist debut at an international art exhibition in 1980s.

The discord results from a rupture between traditional art and new media because the latter, on one hand, features the novelty of advanced digital technology and, on the other hand, includes the irresistible real-time experiences and interactive effects. Today, the new media is still characterized by cross-subject collaboration (as the creation between artists and engineers), new-technological experiments (as the new sensor devices and equipments), the combination of art and commercial development (as art and fashion), theater performance (as the combination of stage space and new media image) and so on. All of them advocate a new coalition between creative thinking and technological potential. Nevertheless, how could the western new media art embodies the oriental creative concept in terms of viewers' participation in the creative work? To be more specific, how can it be possible for viewers to fully understand the artists' contemplation on the meditation experience and their Zenistic concept when their directly interact with the art work by touching them, feeling the space transition, experiencing voice control or the interplay between light and shadow? Hence, the study endeavors to discuss new media in terms of Zenistic expression. It will induce the interaction between new media and oriental Zenism by analyzing the artists' creative thoughts, ways of approach, and content of creation under the oriental and occidental cultural differences.

Key words: Zenism ; New Media Art ; Interactive Installation ; Contemporary Art

壹、緒論

科技數位化的藝術型態是時代的產物，透過電子、網路、程式、電腦等介面的創作，提供創作者與觀者之間新的對話空間，並漸進地改變昔日對藝術作品的觀賞方式。從早期媒體藝術發展至今，這段歷史是跨越舊與新媒體之間的重要橋樑，而結合禪的思維與藝術創作之表現，亦是一種人文思維與文化脈絡的延伸。

一、早期媒體藝術的發展與新媒體趨勢

回顧媒體與藝術的關係，19世紀末的一些關於攝影與通訊技術的發明，開啓人們前所未有的切身經驗，這些改變人類生活的技術包括：1876年貝爾（Alexander Graham Bell）發明電話、愛迪生（Thomas Alva Edison）發明留聲機；1884年出現電影軟片、1888年柯達推出照相機、1889年茲沃里金發明映像管；1890年馬爾

柯尼 (Guglielmo Marconi) 發明無線電通訊技術、1895 年盧米埃兄弟 (Lumiere Brothers) 發明電影、1899 年錄音機與打字機相繼出現，這些產品與發明為當時之媒介傳播打下基礎。特別一提的是照相機出現後，攝影術提供真實的影像紀錄，1887 年英國攝影師邁布里基 (Eadweard Muybridge) 首次使用相機拍攝馬匹的行進動作，以快速曝光拍了一系列馬在跑的照片，目的在於凝住 (Freeze) 運動的刹那，證明馬匹能夠在奔跑時四脚騰空。之後 1909 年意大利詩人馬里內蒂 (Filippo Tommaso Marinetti) 在巴黎《費加羅報》發表《未來主義宣言》，大力讚頌現代工業文明、科學技術使傳統之時間與空間的觀念瞬間改變，主張未來的藝術應當反映現代機器文明、速度、力量和競爭。此時期的藝術觀念從二度空間延伸至三度空間與時間遊移的觀念解析，積極地追求動態性的視覺表徵，包括對於光線、磁力、電力和機械應用等多元媒材的綜合應用。

從 1960 年代藉由電子媒材與類比訊號的嘗試成為當時的實驗形式，沃爾夫·弗斯泰勒 (Wolf Vostell)、白南準 (Nam June Paik) 等人從錄像和影像傳輸為開端，不斷開創電子媒介的視覺語言。直到 1980 年代之後，電腦數位化與新穎的電子媒介相互結合，傑佛瑞·蕭 (Jeffery Shaw)、松墨爾與米諾奴 (Christa Sommerer & Laurent Mignonneau) 等藝術家藉由鍵盤、追蹤器、掃描器、感應器、虛擬實境等數位媒材，藉由多元媒介的創作形式：互動裝置 (Interactive Installation)、互動環境 (Interactive Environment) 和多媒體環境 (Multimedia Environment) 等樣貌，把數位化影像經由擬像 (Simulation)、介面 (Interfaces)、網路空間 (Cyberspace) 整合於藝術的表現型態。

新媒體之「新」，除了功效、表徵與形式上的新穎之意，還包含當下所發生的變動性與創造力。因為它不只是技術和實際介面的總體，同時也是一種圖像、動態、知覺、知識與文化的綜合，里夫·曼諾維基 (Lev Manovich, 2001) 在《新媒體的語言》一書採用「工具—媒介—轉碼」(tool-media-transcoding)、「語言—界面—文化」(language-interface-culture) 等軸線思考訊息再現的意義，對新媒體而言，即是一種文化的運轉與變動。由此探究，從工具、媒介延伸到語言和界面的過程，媒體介面提供了當代藝術更寬闊廣泛的語境，同時包容多元之文化思維和意識型態的範疇。誠如班雅明 (Walter Benjamin) 在其《歷史哲學論綱》(Theses on the Philosophy of History) 中表示：「此刻，任何一個不被視為與己身相關的意象，都將面臨無法挽回的流逝危險；以往歷史學家伴隨心情驚狂而來的喜訊，也許就在一個瞬間而消失殆盡。」(Hannah Arendt, 1999) 他對藝術品即時產生的速度與變異提出警訊，也許是因創作者擁有高度幻想所羈絆，甚至是對於科技的過度渴望而消溶於短暫寄託。雖然，新媒體語言被賦予一種極高期盼而造成審美上的不確定性，但其存在的時代價值與歷史意義，甚至需要更長時間來驗證。

二、禪宗觀念的影響與藝術表現

不同年代對知識和精神豐富的渴求慾望，除了代表精神愉悅和審美態度的滿足外，也逐漸挑戰文化思想與藝術之間的視野。回顧早期中國文人對於禪思想摶注於藝術的表現莫過於禪畫，而後崛起南北宗之說。明代畫家董其昌所創「南北宗」之說，把李思訓和王維視為青綠和水墨畫風的代表宗師。莫是龍《畫說》指出「禪家有南北二宗，于唐時分，畫家亦有南北二宗，亦於唐時分。」至清代方薰所言：「畫分南北宗，亦本禪宗『南頓』、『北漸』之義，頓者概性，漸者成於行也。」（俞崑，1977）這種畫分標榜南宗之文人畫風格，乃出於頓悟。

南北宗原是指佛教史上的宗派，所謂「南頓」、「北漸」即是把「頓悟」和「漸識」（漸悟）作為區別。安世高所傳的小乘禪學，側重於數息行觀的精神修煉，認為達到阿羅漢果位也要累世修行，是主張「漸悟」的一派。支識、支謙所傳的大乘般若學，側重於義解，直探實相本體，後來被看成是近於「頓悟」的一派。東晉時的支道林、道安等人把成佛的修行步驟從「七住」延伸到「十住」，「一住」至「七住」是漸修過程的小頓悟，至「十住」時頓將一切妄惑卻除而得到真正的覺悟。

《大乘理趣六波羅蜜多經》所述「速疾解脫頓悟涅槃」即是頓悟之說，指無須長期按次第修習，一旦把握住真理，即可突然覺悟而成佛。頓悟與漸悟之說在隋唐時期有了新的理解，禪宗主張頓悟說，其餘各宗大都主張漸修。禪宗「南頓北漸」之說，北宗神秀側重漸修，南宗惠能提倡頓悟。惠能認為，由迷到悟只是一念之間，所謂「一剎那間，妄念俱滅，若識自性，一悟即至佛地」，日後逐漸發展成「棒喝」理論，用以破除對方的迷執。再如禪宗的頓悟之說不僅在佛教中影響頗大，對宋明理學也產生很大影響，如朱熹倡導的「一旦豁然貫通」與陸九淵提出「發明本心」即是脫胎於禪宗的頓悟理論。禪宗歷史上的一件著名公案，當五祖傳授心法和衣時，命門下弟子作悟道偈，神秀作偈：「身是菩提樹，心如明鏡台；時時勤拂拭，勿使惹塵埃。」而後惠能的詩偈：「菩提本無樹，明鏡亦非台；本來無一物，何處惹塵埃？」兩人禪語充滿詩性，其概念引發理性的邏輯思維和模糊的原邏輯思維之神秘和互滲。禪的觀念對藝術思維的貢獻不止於邏輯推論，更表現在對原邏輯思維的拓展。

在文藝方面，藝術與禪的思維相互融合模式，例如王維、梁楷、牧溪、玉潤、石谿等人皆是禪藝術的創造者，他們的詩、畫作品無不散發著一種禪意。對於禪畫的看法，陳傳席（2001）在《中國山水畫史》中指出：「受了禪家頓悟精神的支配，猶如禪宗創始人六祖慧能撕經的精神一樣，禪宗繪畫蔑視一切古法，抒發自己自發行爲和直接感受。禪畫力求簡練，不畫名山大川，不畫繁縷復嶺，只畫極普通的山頭，且摒棄細節的刻畫。這一切都和禪宗思想意義相通，所以稱它為禪畫。」早期傳統的創作者把禪宗哲理融入藝術行爲，創造了許多代表性的禪詩、禪畫，甚至影響到日本足利時代的畫風。

將禪的觀念挾注於複合媒材與裝置藝術的表現，當代藝術的創作實踐如賴純純的〈心器〉(1997)(圖1)從〈金剛經〉「若菩薩，有我相、有衆生相、有壽者相，即非菩薩。」體驗而來(高美館，2001)；陳建北的〈對話〉(1990)(圖2)透過具有象徵的材質與宗教符號營造一種冥想空間；徐冰〈911 塵埃〉(2004)(圖3)將禪宗偈語「本來無一物，何處惹塵埃？」(As there is nothing from the first, Where does the dust itself collect?)的文字與灰塵，以此連結於現代人對於災難處境的禪觀思維。上述作品代表了藝術家連結禪思觀念與裝置語彙的印證，兩者相互輝映。而新媒體技術方面，吉姆·坎貝爾(Jim Campbell)的〈影子〉(Shadow, 1993-94)(圖4)，使用LED媒材與佛陀屏幕，讓影像介於真實與虛無之間而不斷產生變化，若有似無的佛像浮現於觀者面前，投影隨即卻又消失。尚馬克·培勒蒂爾(Jean-Marc Pelletier)的〈禪煙〉(Kemuri-mai, 2004)(圖5)，利用Jitter程式、感應器及後製的影像分析，將煙霧與觀者當下呼吸的氣體即時編寫成音頻，最後從煙霧之律動所產生的形狀作為參數，傳送到聲音程式而播放不同節奏的聲音，藉由觀者之吐納與感受而營造出飄渺的冥思場域。

雖然近代西方學者提出「禪」是一種神秘主義(Mysticism)，認為這是一個客觀存在的超越者，或認為宇宙中存在著一種具有超越性的普遍理性(李瑞中，2000)。綜觀上述列舉之藝術家作品，無論是他們所選擇媒材、形式與表現，可看出這是一種觀念的延伸，脫離神秘主義的曖昧差異之後，東西方文化對於禪的藝術詮釋，仍是一種思維，也是一種創作的精神關照。

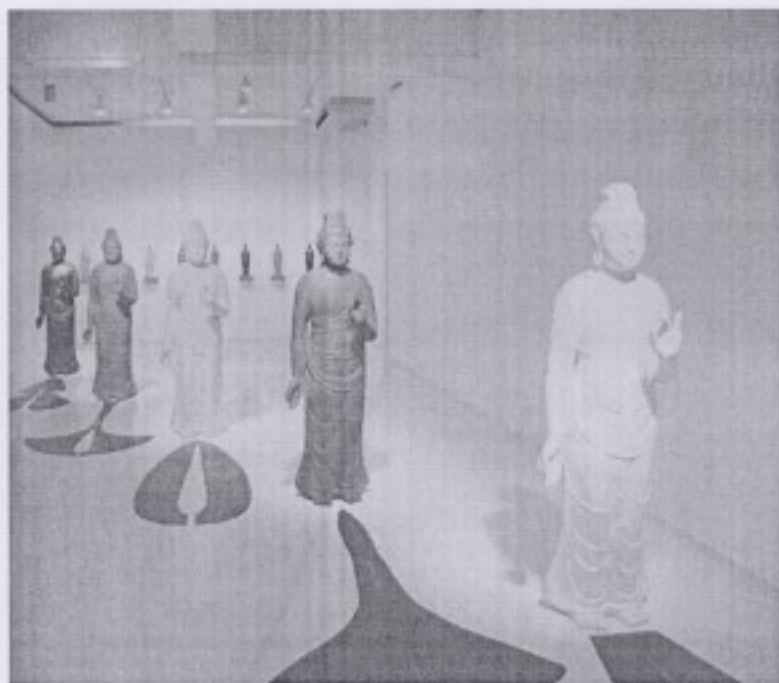


圖1 賴純純，〈心器〉(1997)

圖片來源：姚瑞中。〈台灣裝置藝術〉。台北：木馬文化，2002。



圖 2 陳建北·〈對話〉(1990)

圖片來源：姚瑞中·《台灣裝置藝術》·台北：木馬文化，2002。



圖 3 陳建北·〈911 塵埃〉(2004)

圖片來源：<<http://news.sina.com.cn/c/2004-10-11/17454543402.shtml>>

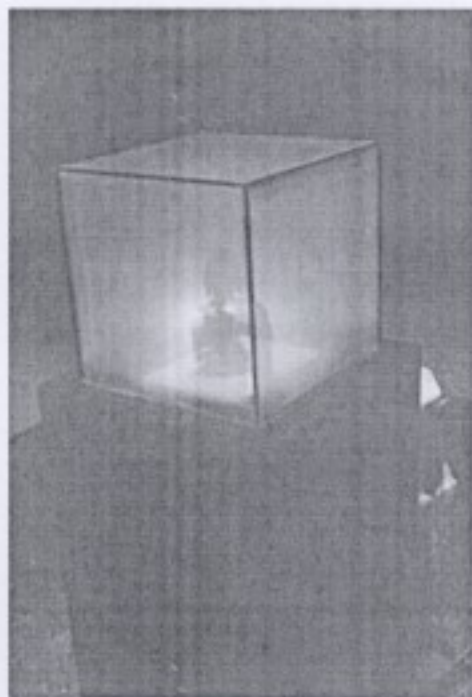


圖 4 吉姆·坎貝爾 (Jim Campbell)·〈影子〉(Shadow, 1993-94)

圖片來源：<<http://www.jimcampbell.tv/IN/INShadow/text.html>>

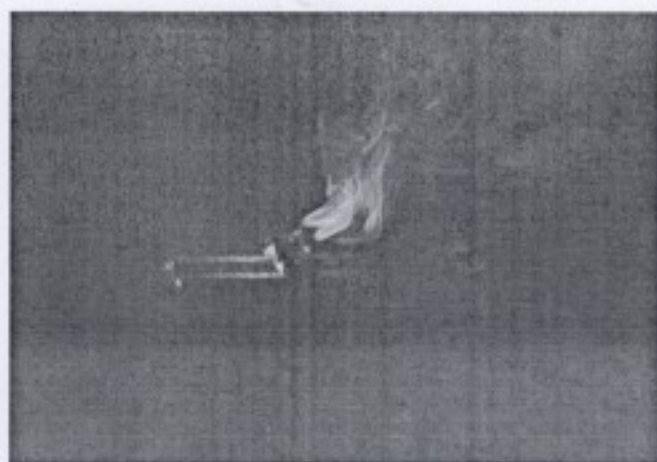


圖 5 尚馬克·培勒蒂爾 (Jean-Marc Pelletier)·〈禪煙〉(Kemuri-mai, 2004)

圖片來源：台灣美術館。〈快感：奧地利電子藝術節 25 年大展〉。台中：台灣美術館，2005。

貳、早期媒體藝術運用禪觀念的例證

承上所述，以禪的觀念當作是心靈領悟之外，若將此與藝術語言相互連結，在審美過程中則帶給觀者不同層次之感通。以近代藝術流派的極簡主義 (Minimalism) 而言，其藝術精神在於去除繁雜的裝飾效果，其講究簡潔、簡約之理念，與禪的「無相」、「無執」等意涵可說相互呼應。再如白南華藉由對於禪觀念的理解，將錄像媒體、傳輸訊號連結於電子媒材，一方面回溯於自身文化背景，另一方面則凸顯媒體應用的新嘗試。如此，藝術多樣化的媒材表現，透過語言、圖像、符號所呈現的視覺效果，與當下禪觀的思維未必可直接解脫智慧，在於突破個人自身省察以及傳達當下的精神存在，仍可作為禪觀念與藝術創作之間的連結。

一、空與悟境的詮釋—約翰·凱吉的作品分析

無論平面、複合媒材或裝置的技巧，藝術從禪學轉化的創作理念通常依創作者的態度決定其內心想法，但經常受到爭議的部分是禪思維過於抽象，如同以「空」來解釋虛實之間的對應關係，無法以科學辯證而衡量。^{註 1}認為「菩提只向心覓」，即是以一種神秘的直覺體驗，把悟性移植到人的心中，從而揭示「瞬間即永恆」與「悟性」的真理，這種所謂「空的智慧」不僅盛行於古代，即使對現代人來說同樣具有啟示精神。(劉大基，2000)

對於空與悟所轉譯在藝術語言上的表現，早期藝術家如約翰·凱吉 (John Cage) 的〈4 分 33 秒〉(1952) 即是一件重要的觀念作品。此作在沒有出現任何樂音、節奏、旋律、和聲的情況下，呈現一種非樂音之靜默的時間結構，過程中，聽眾期待的心情隨著原先期盼聽到真實音樂，瞬間而突然落空。這件作品所開啓之觀念特色，

包括（一）徹底瓦解音樂、解放音符，使作者、演奏者和閱聽人均以身臨其境的方式感受「一切皆無」之狀態。（二）在寂靜時空之下，提出一種由偶發性和不確定性所產生的情境。（三）過程中消彌了創作者的存在，同時讓觀者透過臨場氛圍激發為一種潛在的想像力。約翰·凱吉成為西方前衛藝術的靈魂人物，同時也開啓新媒體觀念的典範，他被藝評家認為是福魯克薩斯（Fluxus）的創舉之作，和東方禪思對於「空」意境的解釋，確實有高度謀合之處。

的確，「空無」是禪學中一個非具象的名詞，在禪宗思想中有所謂「空覺極圓」、「空所空滅」，無論從聲音而來的聞，或知覺性的覺，行、住、坐、臥之中都有空之領悟。依《大正藏》「大聖說空法，為離諸見故，若復見有空，諸佛所不化。」（卷三十）所述，即使一切皆空，但也不等於虛無，既有生滅、得失，它只是說明人們腦中所形成的真實概念。換言之，客觀世界的存在，歸根究底是一種歸零狀態，如果僅從表象存在的思維來理解，這種「空」本身並非是一種什麼都沒有的情況，而是代表另一種精神層次的參悟本質。

二、錄像藝術的媒體禪思—白南準的作品理念

和約翰·凱吉同時期的藝術家白南準，曾在福魯克薩斯藝術節（Fluxus International Festival of New Music）現場表演〈頭部禪機〉（Zen for Head, 1962）（圖6），當羅伯·莫里斯（Robert Morris）唸著詩句：「畫一道筆直的線條」時，白南準以頭部沾上墨汁、蕃茄汁混合的液體，獨自在狹長紙面畫下一道粗獷的黑線，以其行為展現禪的一畫觀念。再如他的〈禪乃為電視〉（Zen for TV, 1963）與〈電影的禪〉（Zen for Film, 1964）則連結電視、電影與禪思之間的影像關係，他以媒體藝術作禪學思維的新平台，凸顯東方文化精神下的視覺表徵。數年後白南準繼而創造〈電視佛陀〉（TV Buddha, 1974）（圖7），在一具古老的佛陀雕像前放置一台電視機，兩者相對位置間架設一台監視器，經由拍攝與直接傳輸的閉路技術，佛像所注視的電視螢幕即為自身畫面。不論是電視陰極射線管之操作或影像同步傳輸的介面，白南準預言了「媒體禪思」的趨勢，並為日後的新媒體藝術留下一道深具文化意涵的伏筆。



圖 6 白南準，〈頭部禪機〉(Zen for Head, 1962)

圖片來源：John G. Hanhardt, "The Worlds of Nam June Paik". New York: Guggenheim Museum, 2000.

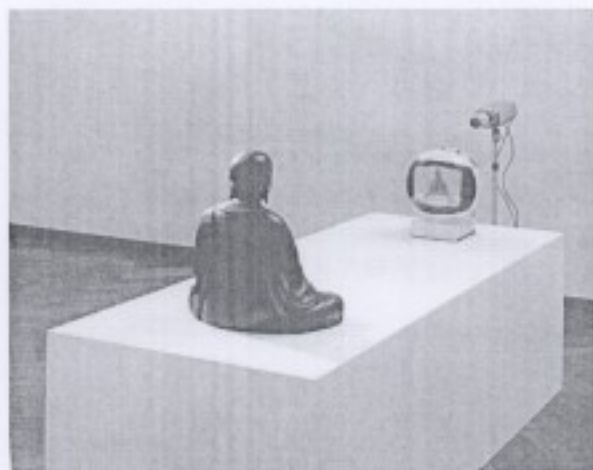


圖 7 白南準，〈電視佛陀〉(TV Buddha, 1974)

圖片來源：John G. Hanhardt, "The Worlds of Nam June Paik". New York: Guggenheim Museum, 2000.

這些例證，說明藝術家分別從身體行為，乃至媒體運用於禪思維與觀念，在歷史脈絡上有著標竿性的表徵。特別是白南準以己身的東方文化背景，在西方物質與媒材運用之對位思考下，同時參酌新技術層次與內化的精神內涵。

然而回顧新媒體藝術發展至今，其學理論述領域與西方體系建構下的史觀謀合仍居於前瞻地位。英國社會學者斯圖亞特·霍爾 (Stuart Hall, 1992) 談論後現代

表徵時指出：「在整個知識體系中，無可辯駁地是以歐洲或西方為中心。」以此推論，無論是媒體資源中心的建構、呈現場域、文本平台等組織結構仍以西方為主體。但是，在自我文化特質的前提下，禪的思維與藝術結合卻早已羅列於所謂前衛的、實驗的、媒體的創作元素之中。尤其在新媒體藝術領域的視野，面對人文禪思與機械複製所製造的影像延伸，似乎已經跨越傳統／創新、西方／東方之間的限制與藩籬。

參、台灣新媒體藝術運用禪觀念的例證

以禪的思維作為創作元素，擴及科技媒體的應用技術，兩者相互結合下的風格如何產生？特別是在台灣當代藝術發展的當下，兩者又產生具有怎樣的意涵？以下分別從〈鏡花水月〉、〈靜心〉、〈心之鯉〉與〈拂拭〉等互動作品，藉此探測創作者以禪思觀念結合電子媒材之手法，以及他們如何賦予科技年代的創新觀點和藝術表徵。

一、吳宗翰、魏德樂、袁廣鳴、孫弘的〈鏡花水月〉

吳宗翰、魏德樂、袁廣鳴、孫弘共同完成的〈鏡花水月〉(2004)(圖8)，水缸中看似裝了八分滿的水，六朵蓮花不斷在水面上漂流移動。使用者手持勺子，可以擾動水面，也可將花撈起，或再倒回水中。創作者汲取「鏡中花，水中月，看似有，卻是無」之文本概念，呈現出鏡花水月、虛幻泡影的意境，並透過影像追蹤系統與動畫視覺處理，轉化為虛擬實境的互動裝置。

從作品架構主體來看，作為顯示與輸入介面的水缸和勺子是觀者可視部分，而隱藏幕後的主要硬體則包括電腦、投影機和3D Tracker。這個3D Tracker 具有磁場發射器(Transmitter)與接收器(Receiver)功能，接收器固定在勺子握把，利用電磁場方式來追蹤物體的位置和角度。電腦和3D Tracker 透過RS-232 連接介面，在系統執行期間電腦不間斷接收3D Tracker 的訊息，同時彙整虛擬環境的內部資料，經過處理後即時產生虛擬景象，再輸出到投影機上，投影機將影像投射在水缸和勺子上，最後呈現出水中撈取蓮花的影音效果。(吳宗翰、魏德樂，2004)

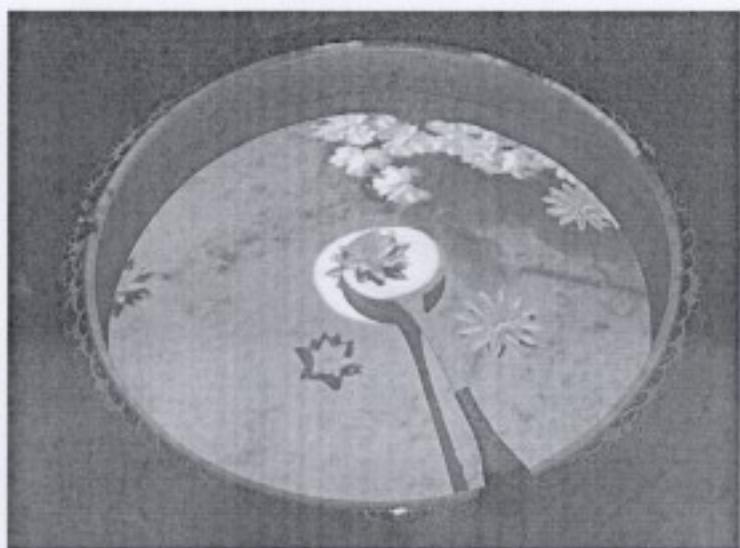


圖 8 吳宗翔、魏德樂、袁廣鳴、孫弘，《鏡花水月》(2004)

圖片來源：「2007 經文與佛像的對話特展」法鼓山（陳永賢攝影）

就詞彙意義而言，「鏡花水月」原作「鏡像水月」，典源來自《遠什大乘要義問答》所述：「火從白石出者，則無香也。風但有觸，而無色也。若非色之物，則異今事。如鏡中像，水中月，見如有色，而無觸等，則非色也。化亦如是，法身亦然。」（卷上，辨法身如幻常光泥洹三義）。典故或見於《大智度論》「解了諸法，如幻，如焰，如水中月，如虛空，如響，如提闍婆城，如夢，如影，如鏡中像，如化。」（卷六）；唐·裴休《唐故左街僧錄內供奉三教談論引駕大德安國寺上座賜紫方袍大達法師元秘塔碑銘》之「水月鏡像，無心去來。徒令後學，瞻仰徘徊。」（《全唐文·卷七四三·裴休》引）；以及《景德傳燈錄》之內文「三界六道，唯自心現，水月鏡像，豈有生滅？」（卷一四，石頭希遷大師）等。

《遠什大乘要義問答》為東晉高僧鳩摩羅什與慧遠的問答錄，書中闡明大乘佛教的教義，如「佛法身者，同於變化，化無四大五根。所以者何？」以此反覆論述有關法身的問題。其中對於「鏡像水月」的釋意是：把地、水、火、風四種元素形成之物稱為「色」，非此四大元素所形成之物則稱為「非色」。凡色，至少具有色、香、味、觸四種感覺之一，例如水有色、味、觸，風則只有觸。若是非色之物，就像鏡中的影像，水中的月亮，看來好像有實體，其實觸摸不著，只是個幻象。

鏡花水月、虛幻泡影的意境，猶如感官之於色、香、味、觸的體驗。透過「來去無礙，通用自由」的凝視與觀想方式，亦如禪宗所稱「戒定慧一體」。何謂戒、定、慧？「戒是外在的，屬於規範性的條目；慧所認知的是理念、理體；定則是去消化與反省的過程。」以一個循環狀態來說，戒（事）→定（境）→慧（理）→戒（事）……等漸次排序。再者，戒定慧一體的意義在於導入「身（身行）→受（感受）→心（心行）→法（法理）」，如此，戒、定、慧三學對於「意」（manas）的解釋從眼（耳、

鼻、舌、身)與色(聲、香、味、觸)為緣而生識之後，以微觀與凝視的審察關係融入生活之中，這層彼此關照之意涵包含「解脫慧」的提示，目的在於解脫智慧、啓悟人生。(釋惠敏，1997)

從「鏡花水月」的內涵延伸至凝視的覺察力，如同以「四念住」(身念住、受念住、心念住、法念住)之方法，察覺出物理、生理、心理的法則。(釋惠敏，1999)之後再依美感原則，觀察自我的生理變化、情緒反應、認識與思考過程及真理確認的覺察，而後轉化為鑑賞能力。因此，(鏡花水月)提供一種有形的凝視覺察，以及無形的觀想作用，參與者透過這種虛擬影像的互動過程，體驗著虛幻泡影的意境。

二、張恬君的〈靜心〉

張恬君的互動裝置〈靜心〉(2006)(圖9)，在簡潔的空間裡置放一張長椅，觀者或坐或臥的姿勢就定位後，投影影像即刻顯示出參訪者身形。螢幕中從上端不斷飄下繽紛花朵，當翩翩落下的花朵落在觀者身上，就像停靠的物件一樣不斷累積。直到觀者抖動手臂或移動身軀時，佇停在身上的花瓣則會紛紛掉落，如同作者所述：「就算是天雨賣花，仍是縈繞於心的煩惱。唯有靜心拋開執著，才能抖落一身的凡塵。」(張恬君，2006)，此概念將觀者靜坐的姿勢與飄落繁花作為比擬，兩者產生一動一靜的寓意關聯。

此作品之互動結構採用 Java 所延伸的 Processing 程式語言，運算過程透過電腦對於像素(pixel)所產生變數值(the value of a variable)的視覺化表現，最後再經由 Webcam 擷取影像而驅動變化結果，投影畫面提供一種即時且鮮明的圖像式反饋，達到觀者與作品之間互動式的感官效果。

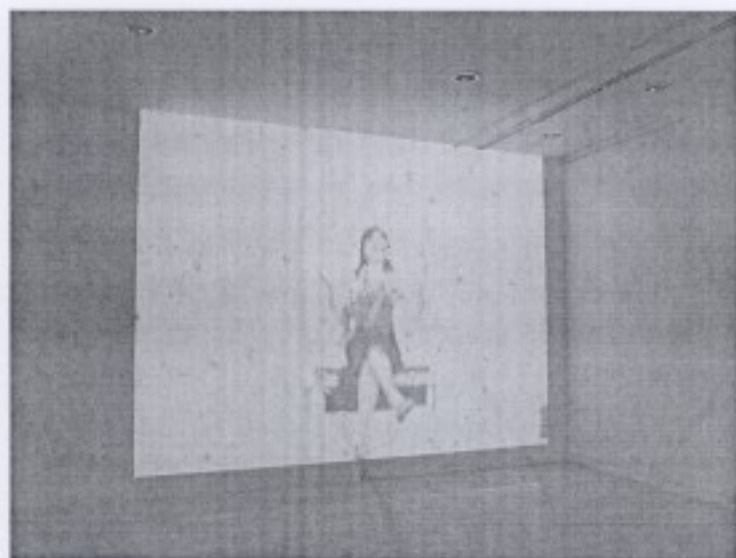


圖9 張恬君，〈靜心〉(2006)

圖片來源：「2006 張恬君個展」鳳甲美術館(陳永賢攝影)

從〈靜心〉作品的內涵可從兩方面來看，第一是關於靜心後的放下（唯有靜心拋開執著，才能抖落一身的凡塵）。以禪的思維來看，因貪慾而招致煩惱憂愁，反復而生的貪慾，明明是無常、不可得而執，無數次地重複於身旁。當拋開過多慾望之後，回歸於自我面貌而得安住。猶如《雜阿含經》所述「觀察五蘊無常，則生厭離。」（卷一）；「離厭，不厭。譬如大力士夫彈指。發聲即滅。」以上所述的生厭離、離厭，均是脫離貪慾後的領悟，而後能夠感到自我知足的詮釋。

第二是「天雨寶花」的生滅比喻。《法華經》中的一段話：「佛告舍利佛，如是妙法，諸佛如來，時乃說之，如優曇鉢華，時一現耳。」生滅的花朵可當作現實生活中的短暫易逝，花朵只是一種意象，藝術家借用自然美與觀者的直覺，造就一種既虛幻又真實的境界，以此延伸至深層之禪意寓言。以寓言作為傳達意念的工具來說，猶如《莊子》所指：「以重言為真，以寓言為廣」可視為一種概念的再現，再以禪思的角度觀察，公案則可視為寓言的另一層解碼。禪公案的內容具有一種弔詭性，成中英（1994）認為，禪之疑惑論證主要是源於對話者故意把公案之表面語意，以及深層存有論的指涉關係予以切斷，致使表面意義的語言失去了本身的架構，但是，禪之語言卻因而獲得新意義。由此延伸，張恬君的〈靜心〉提供之理念，指涉一種超驗的、絕對的、永恆的價值，不僅存在於作品外部的意象，同時也提供參與者在體驗觀想時，有了新的重組意義。

三、黃心健的〈心之鯉〉與〈拂拭〉

黃心健的〈心之鯉〉（2004）（圖 10）從《金剛經》「應無所住，而生其心」發想而來，藉由牆面投影的「心」字所組成的虛擬鯉魚，跟隨觀者的影子而穿梭，當觀者的影子擋住牠的去向時，鯉魚避開後順勢地轉折到白色的畫面，過了若干時間，牠又會逐漸消失而歸於虛空。這隻鯉魚的意涵宛如分割成亮與暗兩個部分，誠如作者所說：「體外的世界，反而比體內的世界容易瞭解。我們用語言與工具，可以對這外在的世界作精密的量測與操縱；然而，對體內的世界，卻像是一個黑洞，隱藏著各種情緒與記憶，這些，只能透過當事者自發性的抒發，藉由有限的語言文字，才得以與外界溝通。」（黃心健創作自述）虛擬鯉魚和牆上投影，兩者意味著體外世界的意識層和體內世界的黑洞，以此表達各種顯性與隱性的情緒與記憶溝通，亦如禪意棒喝般的覺醒暗示。

黃心健另一件作品〈拂拭〉（Dusting, 2007）（圖 11），創作概念從神秀與慧能的兩個偈子得來靈感：「身是菩提樹，心如明鏡台，時時勤拂拭，勿使惹塵埃。菩提本無樹，明鏡亦非台，本來無一物，何處惹塵埃。」裝置呈現的效果是經由觀者的手輕輕掠過桌面，如果桌面沒有文字，則會隨即產生文字，逐漸浮現《法華經》的經文字句慢慢接連成串，文字串成像是一條游動的魚，牠會閃躲觀者的手、或是

其他障礙物，最後消失在桌面上。若桌面已經出現文字，文字會因觀者的擦拭動作而漸消失。

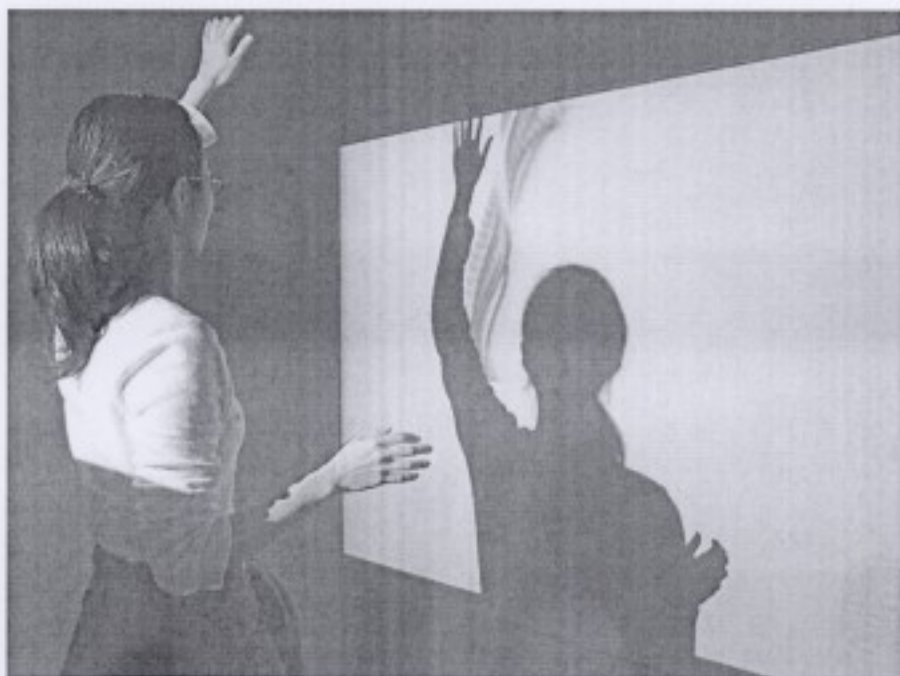


圖 10 黃心健·〈心之鏡〉(2004)

圖片來源：「經文與佛像的對話特展」法鼓山(陳永賢攝影)



圖 11 黃心健·〈拂拭〉(Dusting, 2007)

圖片來源：創作者提供

將禪思意念轉化為媒體裝置的空間，其身歷其境的臨場感與即時回饋作用，已經超越二度空間的一般想像。上述黃心健的作品特色，可以窺探出他同時訴說臨場空間與觀者短暫棲居的互動狀態，藉由剎那的時間性給予觀者一種當下對於禪思維的體驗與辯證。這種即時提供體驗式的參悟，直接取材於經文和偈語，無論對於禪的漸悟或頓悟感受，作品呈現都給予觀者新穎的、愉悅的、互動的、啟發性的新視野。

對照於美學觀點之「感情移入」、「情景交融」的藝術意境，也和作品所提供偈語悟境有著異曲同工之處。人們如何參與禪的悟境？釋聖嚴（1992）認為：「悟境確實是有的，但卻不是以企求心、好奇心所能得到的，以猜測和想像來揣摩悟境，則永遠無法瞭解悟境是什麼。『開口即錯，動念即乖』，若用思想、知識、語言、文字、邏輯推理等方法來解釋說明悟境，便與悟境的本身，越離越遠。」此外，他曾以〈十牛圖〉為例並指出：「開悟層次，乃是因人而異的……有的人初悟與再悟，就有了淺與深的感受。」禪的悟境是人人皆有，而對於物我的情景交融，則是初禪五支「覺、觀、喜、樂、心」的體驗，相通於藝術審美的感知。如此來看黃心健的作品，可以說在藝術鑑賞過程中，他提供一種禪思悟境的空間，進而在主客體合一的氛圍下，釋放出自由的參悟機緣。

肆、結論

整體而言，近年來新媒體藝術發展下所凸顯的特色，重點包括兩部分。首先，以媒介物件的傳達與溝通來看，舊媒體的單向性傳播與新媒體的交互性相比，後者明顯將參與者沈浸式的體驗過程緊密連結，並使其互動過程形成一種流變的、開放的、動態的和體驗的審美狀態。互動性是新媒體作品的顯性特徵，其呈現機制直接促使電子媒介成為重要元素，猶如史蒂芬·威爾森（Stephen Wilson, 1993）所指：「互動，代表著控制者、瀏覽介面與觀眾之間，彼此產生作用、影響事件過程與當下發生的狀態。」換言之，當觀眾介入互動機制，表面上是藉由互動效果達到作品解釋之目的，實際上卻消彌創作者與觀者之間的距離，更讓解讀藝術的意識轉嫁於參與者的行為過程。

其次，以新媒體藝術的當代性來看，創作形式逐漸走入跨領域的結盟趨勢，組合類型如（一）綜合性的跨領域合作：如「爆破理論」（Blast theory）、「藝術+公司」（Art + Com）等團體結合藝術家與工程師之間的創作組合；（二）新技術的實驗風格：如拉法耶·羅札諾－漢墨（Rafael Lozano-Hemmer）以嶄新的感應裝置與器材設備，創造新穎的視覺與肢體感受；（三）藝術與商業的結合開發：如倫敦 Night Club 邀請藝術家創造兼具藝術魅力與流行時尚的影音效果；（四）表演形式與呈現：如 DV8 劇場、Ultima Vez 舞團將舞台空間與新媒體影像相互融合等，均在創作思維

和技術潛能之間形成了良性的合作關係。

的確，在新媒體藝術範疇中賦予禪的思維，成爲一種文化意識中的表現方式。如前所述，審美機制的經驗，特別是從觀者參與過程中所經歷的轉化來說，如觸摸、空間移動、聲音控制、光影感應等互動介面，移轉至屬於自我內化的參悟，即是科技與數位化之後的藝術語言。禪的思維與新媒體藝術，兩者相互結合而有新的形式，在當代藝術潮流下的可能結構與未來展望，歸納如下。

一、新媒體藝術運用禪的思維在當代藝術潮流下之可能結構

無論東西方的文藝脈絡，文化意識融入藝術表現是長久以來方式之一；然而，面對電子媒材、電腦技術的時代新產物，透過禪思觀念的具體實踐又可成爲當代潮流下的另一種意涵，其精神結構包括：

（一）數位革命導致一種新的全球共存關係，也引發人們對媒體與藝術空間的新思考。創造新技術、利用新技術，是當代藝術發展的必然趨勢，而與日遽增的科技產品，以及透過電訊傳輸和結合電腦創作的媒體藝術，與科學或生物學結合理論的建構，尤其是文化意識與信仰思維的內涵更顯重要。

（二）姑且不論新媒體藝術的發展始於何地，在西方經驗歷程的圓周閉合之後，開創在地文化與本地精神的自我特色，繼而創造屬於己身的新圓起點。換言之，在快速變遷的電腦媒介裡，挹注更準確的技術、創意之外，新媒體藝術與禪思觀念的美感結合，即是另一種借古創新的表現方式。

（三）在新媒體藝術與禪思觀念結合，除了再次燃起人們對電子科技的憧憬之情，即刻處在於一個他我異質的存在意識。在此兩者之間，一爲不可或缺的媒材介面，一者爲內化的禪思維體驗。顯然，數位技術和影像奇觀在此扮演相當重要的角色，它隨著創作者的詮釋手法，進而帶領觀者進入這層被營造出來的文化思考與氛圍之中。

二、賦予禪的思維於新媒體藝術之創作實踐與展望

創作者藉由科技與新媒體形式所呈現的禪思語彙，其共通特質即在於空間營造與觀者互動之間的體驗，而表述的實踐與展望在於：

（一）新媒體藝術透過空間場域營造一場虛擬幻影與真實介入的場域，而人們對於空間的感受則如同巴舍拉（Gaston Bachelard, 1994）所述：「空間不是一個均質空洞的範圍，而是一個充滿了質量甚至幻想的空間。是我們最初感知的、夢中的

空間，我們的情感原本就屬於其中一部分的空間。」假如將此場所轉移至新媒體的擬像空間，創作者提供一種禪悟的冥思場域，亦即包容身體存在所投射的虛擬影像，兩者在於感受真實和變異瞬間之時，共同被感知於禪觀的審美場域中。

(二) 觀者進入展場空間之後，在數位幻影與禪思維的意境體驗中，藉此找尋一種似近非遠的科技時光。此刻，參與者的感受、生活與禪思的距離不再是遙不可及，而感受禪思觀念不僅是傳統文化的一部份，也可成為「當下即是」的體驗美學。

(三) 號稱科技之島的台灣，有著多元的先進技術開發與產業，未來若能結合更多科技產業與藝術創作結合，以及開發專業工程師與創作者之間的共創平台，對台灣新媒體藝術的發展將有具體且實質的助益。

總而言之，新媒體運用的內容即是藝術實踐的根本，它同時建構於優美的傳統與既有的意識形態之中，同時也需要在文化意涵中找尋新的美學方向。換言之，形塑台灣當代藝術於此雙邊的藝術能量，不僅是新媒體與科技間的關係，亦同時指涉藝術創作在文化意識上的覺醒。

參考文獻

- 成中英 (1990)。〈禪的詭論和邏輯〉，《中華佛學學報》第三期。台北：中華佛學研究所，頁 185-207。
- 吳宗翰、魏德樂 (2004)。〈鏡花水月--虛擬實境互動裝置〉，《國巨科技藝術國際學術研討會》。台北：國巨文教基金會，頁158-163。
- 俞崑 (1977)。《中國畫論類編》。台北：華正書局。
- 高美館 (2001)。《心靈再現：台灣女性當代藝術展》。高雄：高雄市立美術館。
- 張恬君 (2006)。《萬里雲天》。台北：鳳甲美術館。
- 陳傳席 (2001)。《中國山水畫史》(修訂版)。天津：人民美術出版社。
- 黃河濤 (1997)。《禪與中國藝術精神的嬗變》。台北：正中書局。
- 鈴木大拙著，孟祥森譯 (2000)。《禪學隨筆》。台北：志文出版社。
- 鈴木大拙著，劉大悲譯 (2000)。《禪與生活》。台北：志文出版社。
- 潘平、明立志編 (2005)。《胡適說禪》。台北：九儀出版社。
- 國立台灣美術館 (2005)。《快感：奧地利電子藝術節 25 年大展》。台中：國立台灣

- 美術館。
- 釋聖嚴（1992）。《禪的體驗》。台北：東初出版社。
- 釋惠敏（1997）。《禪定與生活》。台北：西蓮淨院出版社。
- 釋惠敏（1999）。《戒律與禪法》。台北：法鼓文化公司。
- Ales Erjavec 著，胡菊蘭、張雲鵬譯（2003）。《圖像時代》（*Toward the Image*），吉林：吉林人民出版社。
- Baudrillard, Jean 著，洪凌譯（1998）。《擬仿物與擬像》（*Simulacres Et Simulation*），台北：時報出版。
- Bachelard, Gaston 著，龔卓軍、王靜慧譯（2003）。《空間詩學》。台北：張老師文化。
- Bourdieu, Pierre 著，林志明譯（2002）。《布赫迪厄論電視》（*Sur la television*）。台北：麥田出版社。
- CBETA 電子佛典。
- D.Nicholas Mirzoeff 著，陳芸芸譯（2004）。《視覺文化導論》。台北：韋伯文化。
- Mitchell, William 著，吳啓迪等譯（2001）。《伊托邦—數字時代的城市生活》。上海：上海科技教育出版社。
- McLuhan, Marshall 著，何道寬譯（2000）。《理解媒介》。北京：商務印書館。
- McLuhan, Marshall 著，何道寬譯（2000）。《麥克盧漢精粹》。南京：南京大學。
- Stevenson, Nick，王文斌譯（2001）。《認識媒介文化》。北京：商務印書館。
- Wilson Ross, Nancy 著，徐進夫譯（1994）。《禪的世界》。台北：志文出版社。
- Arendt, Hannah.(Ed), W.Theses, Benjamin.(1999). *On the Philosophy of History*, London: Pimlico.
- Austin, James H. (1999). *Zen and the Brain: Toward an Understanding of Meditation and Consciousness*. London, The MIT Press.
- Bodri, William and Lee, Shu-Mei. (1998). *Twenty-Five Doors to Mediation*. York Beach: Samuel Weiser INC.
- Bussman, K. & Matgner, F. (1993). *Nam June Paik*. Venice: German Pa.
- Christmas, Humphreys. (1988). *Zen Buddhism, Australia*, Allen & Unwin Paperbacks Ltd.
- Cleary, Thomas. 1986. *Zen Essays by Dogen*. Honolulu: University of Hawaii Press.
- Ernest, Wood. (1977). *Zen Dictionary*. New York: Philosophical Library.
- Evans, Jessica & Hall, Stuart. (1999). *Visual culture, the Reader*. London: SAGE Publications Ltd.
- Hansen, Mark B. N. (2006). *New Philosophy for New Media*, Cambridge: The MIT Press.
- Hall, Stuart. (1992). *The Question and Cultural Identity*. Cambridge: Polity Press.
- Manovich, Lev. (2001). *The Language of New Media*, Cambridge: The MIT Press.
- Rexroth, Kenneth. (1981). *The Buddhist Writings of Lafcadio Hearn*. London: Biddles Ltd.

Suzuki, D.T. (1972). *The Zen Doctrine of No- Mind*. York Beach: Samel Weiser Inc.

Suzuki, D.T. (1977). *Essays in Zen Buddhism*. London: Rider and Gomapny Ltd.

Suzuki, D.T. (1977). *Zen Buddhism*. London: Rider & Company Ltd.

Suzuki, D.T. (1990). *The Zen Doctrine of no Mind*. London: The Guernsey Press Ltd.

Tribe, Mark & Jana, Reena. (2006). *New Media Art*, Taschen GmbH.

Wood, Ernest. (1957). *Zen Dictionary*. England: Penguin Books Ltd.

<<http://news.sina.com.cn/c/2004-10-11/17454543402.shtml>>[2007-04-29] ◦

<<http://www.jimcampbell.tv/IN/INShadow/text.html>>[2007-05-14] ◦

<http://www.digiarts.org.tw/Resource/LeaRec_detail.aspx?LeaRecDBID=14> [2007-05-13] ◦

<<http://www.fundacion.telefonica.com/at/rlh/index.html>> [2006-05-05] ◦

第三章、數位奇觀與新互動倫理

—當下數位藝術的發展及可能性

撰文/陳永賢教授

本文出處：

陳永賢·〈數位奇觀與新互動倫理—當下數位藝術的發展及可能性〉·《藝術家雜誌》(2007年9月,第388期):162-167·(藝術家雜誌9月專輯)

壹、視覺文化的變異

隨著現代科學與電腦的變革，數位媒介的日新月異迫使影像文化全面滲透到日常生活。回顧普普藝術盛行的年代，李查·漢彌頓(Richard Hamilton)以拼貼手法預告下一個世代的未來性。他在1956年的〈是什麼使今日的家變得如此不同，如此吸引人？〉，這件英國最具典型代表的拼貼畫作，濃縮了現代消費文化特徵之外，畫面上出現電視、卡帶式錄音機等媒體物件，以及天花板上的月球圖象，似乎預告日後科技介入生活的普遍性。



李查·漢彌頓 是什麼使今日的家變得如此不同，如此吸引人？ 拼貼 26×25cm 1956

的確，當今家庭從類比進入數位化設備的現象已經相當普及，而數位化的藝術表現也成為當代藝術中重要的一環。

然而，平面影像的異度空間也是創作者亟欲探求的目標，藝術歷史的走向從1960年代未來主義、達達主義、動力藝術，進而從觀念藝術、偶發藝術、行為藝術與數位化媒體結合，藝術創作者表現新思維與新經驗，並且回應科學的感知和意識。杜象具啓發性的〈下樓梯的裸女〉，讓人聯想到現今機械人或人工智慧的研究，其中跨越時間因素的視覺暫留也成為後輩追尋的指標。在大衛·洛克白 (David Rokeby) 〈廉價模仿〉的作品中，藉由數位化的影像模組，刻意回應杜象對於斷片、合成、動力等拼貼方式的運動時間。因此，視覺藝術從二度空間跨越到三度或四度空間的向性迴轉，藉由數位機器 (電腦/計算機) 的漫遊程式，經過連結、融入、互動、轉化、出現等五個階段，以此改變視覺上聚合離散的焦點。



大衛·洛克白 廉價模仿 電腦、錄影機、錄影數位化機、客製化軟體 2002

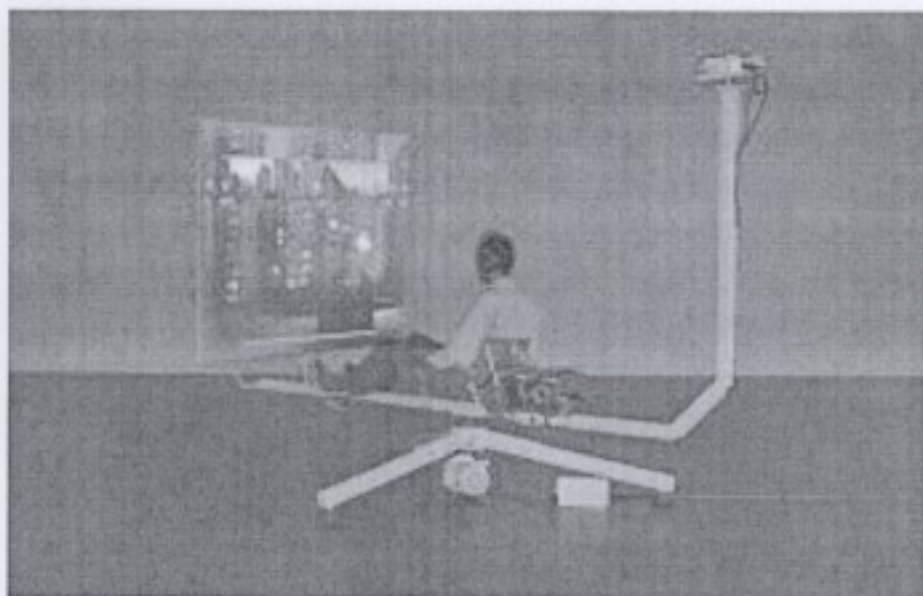
無可諱言，藝術創作者透過觀景窗或電腦視窗，觀眾透過螢幕或一層「超薄」的投影影像，你我看到電子傳輸後的視覺表徵，促使一切視覺文化的符碼正不斷被改寫。而數位化的藝術特質在於連結性與互動性，強調作品與觀眾之間的彼此關係，轉化後的影像出現全新的思維與經驗。種種形式的視覺差異，無不表述這是當下數位藝術所彰顯新媒介、新實驗、新思維與新體驗的一環。

貳、觀看方式的變遷

拉康 (Jacques Lacan) 早年曾提出「鏡像階段」理論，他改變柏拉圖的「洞

穴比喻」之說。柏拉圖以執迷於洞穴幻影的囚徒所逃向真實光源之過程，是藉由視覺理性與自我認知之眼而獲得的真理啓示。在拉康的想法，鏡像階段的嬰兒對鏡中之「我」的形象確認，終究不是一種誤認，而是自我異化的完形概念，「觀看」是對於自我形構與自我導引的指示作用。之後，拉康進一步說明凝視的原則，他指出：「凝視不僅是主體對物體或他者的觀看，而且是做爲慾望對象的他者對主體之注視，這種彼此相互作用，是主體在異形之他者於凝視過程中的定位。」

過去的經驗，我們憑藉五官和外界產生知覺關係，如今眼睛所觀察的部分，透過數位化科技的輔助，不單可以看到 360 度的景觀，更可以看見世界不同位置所發生的事情，如遠距通訊、網路視訊、虛擬實境等工具去傳播動態影像。因而數位化影像的凝視作用悄然驟變，馬尼克斯·德·奈斯 (Marnix de Nijs) 的〈北京加速器〉即是一例，安裝於旋轉座椅與移動螢幕的速度控制，彼此共構環場視野與新穎的觀看方式。以科技數位化的創作而言，無論在審美態度和媒材應用的關聯性，創作者使用數位媒體的求變態度不斷力求突破新局，對拉康凝視之本體論已逐漸產生衝擊。尤其數位化所營造的虛擬空間和互動機制等概念，漸進地改變昔日對藝術品的觀賞方式。如此的視覺感受，無疑已脫離對傳統作品的觀看方式，對觀者沉浸式的回饋與經驗來說，已經演變爲新的知覺感受。



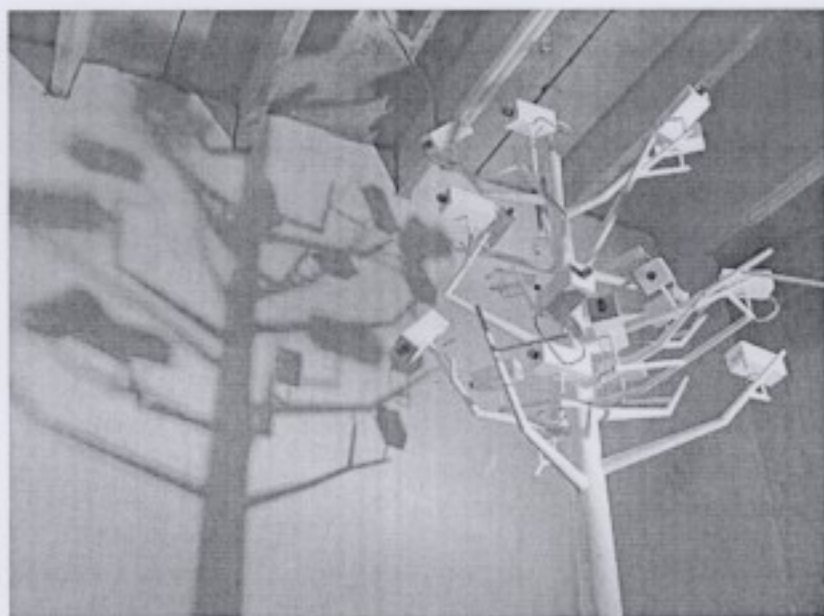
馬尼克斯·德·奈斯 北京加速器 機械零件、電腦、投影機、電器、感應器、螢幕、鋼構、競賽椅
375×375×220cm 2006

參、互動：在場與不在場

以媒介物件的傳達與溝通來看，單向性視覺傳播與媒體的交互性相比，後者明顯將審美概念與參與者緊密連結，並使參與作品的互動過程形成一種流變的、開放

的、動態的和體驗的美感認知。假如以具有互動性的特質為判斷，那麼，藝術呈現的執行機制亦直接促使數位媒介成為重要元素，如史蒂芬·威爾森(Stephen Wilson)所指：「互動，代表著控制者、瀏覽介面與觀眾之間，彼此產生作用、影響事件過程與當下發生的狀態。」如此，當觀者介入互動機制，表面上是藉由互動效果達到作品解釋之目的，實際上卻消弭創作者的主觀元素，使解讀藝術的意識轉嫁於參與者行為之上。

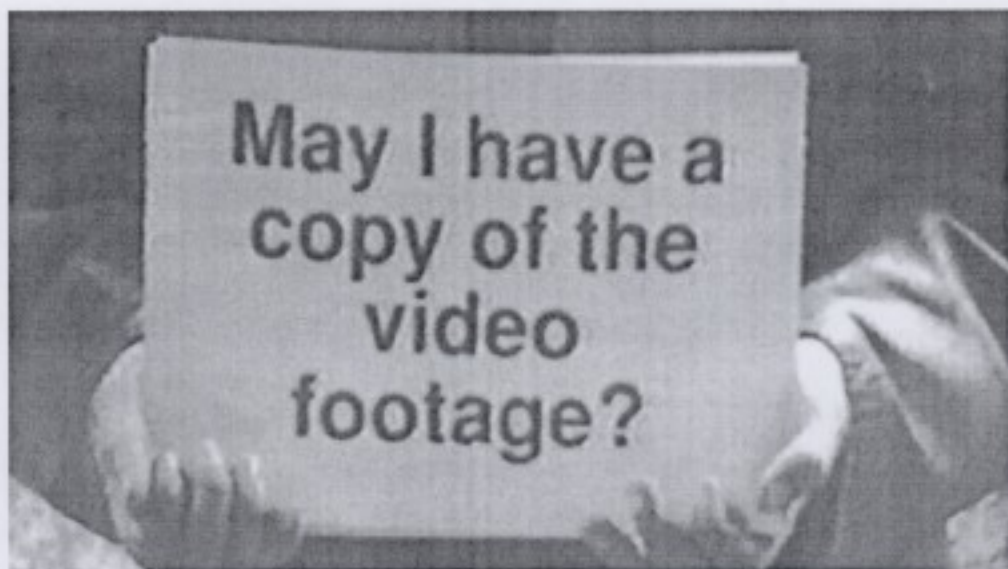
然而，當下數位化、媒體化的生活圈無疑就是注視／監視的雙重關係，你是否注意到身旁眾多監視系統圍繞於生活周遭？這種隱藏背後的觀看方式，正如英國哲學家邊沁(Jeremy Bentham)設計的「全景敞視監獄」，它不僅是個透明的牢籠，也是個完美的規訓機構模型，更是一種精神對精神的權力運作體現。傅柯(Michel Foucault)曾對此延伸至主體被建構的機制，以此揭示視覺中心的理性話語中既寬敞又隱蔽的非理性景觀。傅柯在 1977 年接受一次題為「權力的眼睛」的訪談中談到：「邊沁所夢想的是一個透明的社會，一個沒有任何黑暗區域的社會，每一部分都可見而易讀……。」對於數位媒體的縮影而言，啟動監視權力的背後，仍是另一個異次元空間的營造，猶如麥克魯漢在《理解媒介》中所述：「人們對媒介的性質沒有覺察，正是由於注意媒介內容的同時忽略了媒介的形式……，對媒介影響潛意識的溫順的接受，使媒介成為囚禁其使用者的無牆監獄。」



亞歷山大·阿瑞洽 猜忌的花園 錄像裝置 2004

因而，數位化的互動呈現形成一個既有趣又詭譎的處境，同時表述注視／監視、在場／不在場的曖昧關係。例如亞歷山大·阿瑞洽(Alexandra Arrechea)〈猜忌的花園〉，指涉電子產物的監看以及人際關係疏離，成為現代紛亂社會的特徵；丹尼·布鮑塞(Denis Beaubois)〈在失憶時這城市會想起〉即以行動挑戰控制者的監

視系統；斯達普拉聲響系統（Staalplaat Soundsystem）（台灣製造）讓觀者無論處於在場或不在場，均可撥打手機號碼而控制任何可以發出聲響的組件。於此，透過里夫·曼諾維基（Lev Manovich）的數位化呈現、模組化、自動化、可變易性、貫通式編碼等五項原則，我們深刻體驗到數位互動的程序。然而在穿越時空場域下的身體狀態，無論在於注視或監視、在場或不在場，表面上回到「全景敞視監獄」的凝視氣氛，實際上卻透過這種彼此交叉監看之過程而啟動新的對話美學，同時又形構另一層次的符碼解釋。



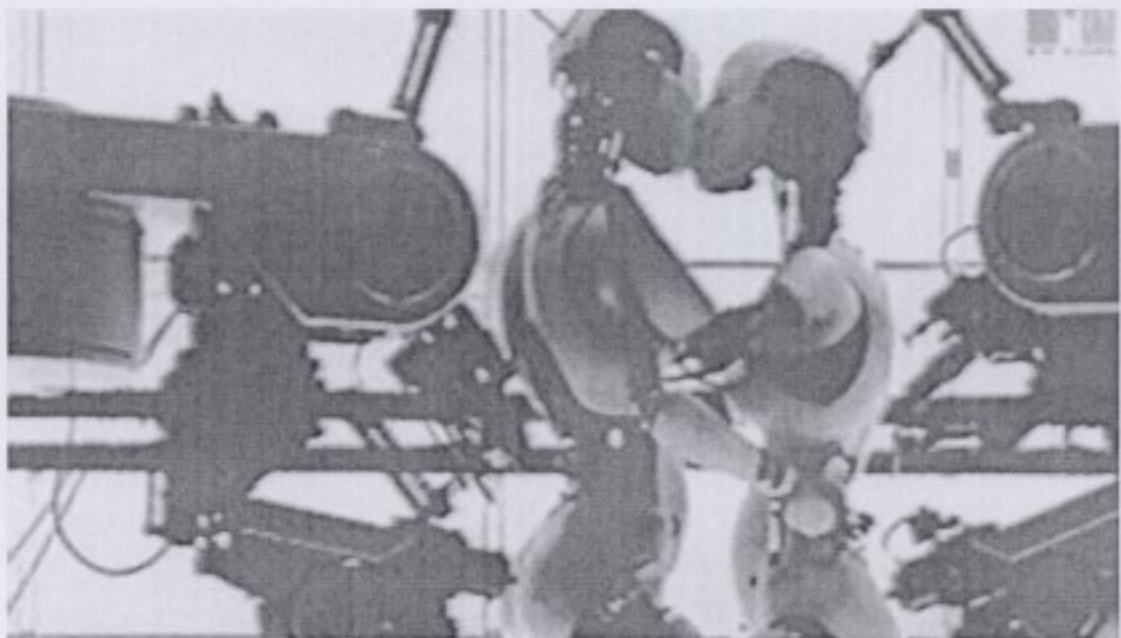
丹尼·布龍奎 在失憶時這城市會想起 錄影／行為表演 5分鐘 1997



斯達普拉聲響系統 台灣製造 廢棄家電及工具、玩具樂器、玻璃聲響系統、電腦、GSM 數據機、max/msp/flash 控制軟體 2005-2007



薩克利·利柏曼 繪 客製化軟體、投影、墨水、繪紙 2005



克里斯·康寧漢 全都充滿愛 DVD 投影 4分08秒 1996



池田亮司 數據原型 翻畫視覺藝術裝置 2007

肆、數位藝術的奇觀性

回頭看看霍爾班 (Hans Holbein, 1497-1543) 的〈使節〉, 畫面描繪兩位年輕的使節與主教, 看似平凡的寫實肖像畫, 不過從背景擺飾物件中卻反應出幾何、算學、天文、音樂等隱喻元素。再仔細觀察畫面下方的馬賽克地磚中央, 出現一具虛幻的骷髏形狀, 形塑此畫一種神祕而奇觀的景象。視覺的奇觀性自然吸引觀者的目光, 同時在虛實幻象的推測過程, 迷宮探尋般的回饋心理亦是一股無法阻擋的魅力。



霍爾班 使節 油彩、橡木 207×209.5cm 1533 倫敦國立美術館藏

在薩克利·利柏曼 (Zachary Lieberman) (繪) 作品中，觀者可以拆解、重組已經形繪的圖案；克里斯·康寧漢 (Chris Cunningham) (全都充滿愛) 製造一齣未來式機器人的情愛憧憬；而池田亮司 (Ryoji Ikeda) (數據原型) 則擴大物理和數學的計算數據為壯闊頻率與浩瀚視覺；喬治·拉格迪 (充滿記憶的口袋) 收集四面八方所提供掃描的物件，擴增為聚沙成塔般的視覺編碼。賀維格·懷瑟 (Herwig Weiser) (z 神探測器/zII 版) 以磁性金屬微粒的相斥與相吸原理，以及收音麥克風、電磁感應器於粉末化土壤中的編碼程序，製造一座可變式的磁場奇觀。



喬治·拉格迪 充滿記憶的口袋 掃描機、電腦、相機、投影機、客製化軟體、網路連線 2001-



賀維格·懷瑟 z 神探測器/zII 版 樹脂玻璃結構、硬體磨砂、磁力液體、電器組件、音響系統、電腦、178DC 電磁體、178LED 燈架及其他 42×77×91cm 2005

當數位藝術的奇觀性成爲一個微型縮影，豈不可建構出一套挑戰視覺驚奇的表徵？德博（Guy Debord）在《奇觀社會》曾說：「在現代生產條件蔓延的社會中，其整個生活表現爲一種巨大的奇觀積聚。曾經直接地存在的所有一切，現在都變成了純粹的表徵。」表徵即圖象，今日的圖象已經成爲生活中的物質性力量，視覺圖象不再只是反應和溝通，尤其數位化圖象（虛擬、錯置、表述）被建構、折射或扭曲，其深層意義是，技術化的視覺經驗透過機器編碼程序，而形成機器拓展後的視覺經驗。這也意味著，不再有本質與現象、真實與表象之分，甚至有超真實（Hyperreality）意味。布希亞（Jean Baudrillard）以「擬真」（Simulation）爲例談到：「不再有存在和表象的鏡像，不再有現實和現實之概念的鏡像。不再有想像性的共存，做爲擬像之向度是基因的微型化。真實從微型化細胞、母體和記憶庫，以及控制模型中被製造出來，並且被無數次複製出來。」

綜合上述，當下數位藝術的可能性充滿著變數，冀求掙脫傳統卻也難只侷限於技術層面，涵蓋於觀念意識之下的創作符碼可能引起一陣喧騰，也可能稍縱即逝。對於數位藝術的延續性發展爲何？最後以班雅明（Walter Benjamin）的話語做爲結尾，他說：「此刻，任何一個不被視爲與己身相關的意象，都將面臨無法挽回的流逝危險；以往歷史學家伴隨心情驚狂而來的喜訊，也許就在一瞬間而消失殆盡。」

第一章、Processing 程式概論

【Webcam 與多媒體創作】工作坊

大綱

- 1-1. Processing 程式簡介
- 1-2. Processing 的編輯環境
- 1-3. Sketch 專案
- 1-4. Applet 匯出
- 1-5. Application 匯出
- 1-6. 全螢幕顯示
- 1-7. 影像座標系統
- 1-8. 影像顏色系統
- 1-9. 主程式架構 (Programming Modes)
- 1-10. 影像產生模式 (Rendering Modes)
- 1-11. 基本程式結構
- 1-12. 進階程式結構
- 1-13. 程式庫 (Processing API)
- 1-14. 延伸閱讀

1-1. Processing 程式簡介

網址: <http://processing.org/>

Processing 是一個開放原始碼的程式語言及開發環境，提供給想要對影像、動畫、聲音進程式編輯的工作者。當然，學生、藝術家、設計師、建築師、研究員以及有興趣的人，也可以用此程式來學習，開發原型及製作。

開發這套軟體的目的，是為了教導學習者一些以視覺呈現為主的電腦程式基礎，並且將這套軟體看作是一個軟體的描繪本，以及專業的製作工具。

Processing 是由一群藝術家及設計師所開發的，在相同的領域上，是其他商業性軟體開發工具之外，作為另一個可以選擇的工具。

Beta 測試版軟體 1.0 版於 2005 年 4 月 20 日發布，並可在本網站下載，錯誤的修正會一直進行著，直到 1.0 版正式發布。Processing 可以免費地下載，支援的平台有 Linux、Mac OS X 及 Windows。

Processing 是一個開放的專案計畫，由 Ben Fry (Broad Institute) 及 Casey Reas (UCLA Design | Media Arts) 發起，並且是由 MIT Media Lab 的 Aesthetics and Computation Group 的構想發展出來的。

目前最新的版本是 0125 BETA

下載網址: <http://processing.org/download/index.html>

1-2. Processing 的編輯環境

Processing 程式編輯視窗





1-3. Sketch 專案

所有在 processing 開啓的一個專案都稱爲一個 sketch 專案，每個 sketch 專案都有一個和 sketch 專案名稱相同的目錄，並在此目錄中產生一個 sketch 主程式。例如：若專案名稱爲"Sketch_123"，則 sketch 目錄名稱就是"Sketch_123"，而 sketch 主程式名稱就是"Sketch_123.pde"。

sketch 專案中若有使用影像、圖片、聲音、字型檔案時，要將檔案放在 sketch 目錄下的"data"目錄中。

當執行匯出功能時，系統會將 data 目錄下的所有檔案，匯出到一個壓縮檔。例如："Sketch_123.jar"。

原則上，sketch 專案的檔案都會放在 Processing 的安裝目錄下，若要將 sketch 專案的檔案放在其他的目錄的話，可利用"File"功能選單中的"Preferences"功能來設定。

另外，可以同時有多個程式檔案放在同一個 sketch 目錄下。可以放置 Processing 原始檔（附檔名是.pde），以及 Java 原始檔（附檔名是.java）。

1-4. Applet 匯出

若要讓 **sketch** 程式可以在網頁中執行的話，利用 **Export** 功能，系統會將 **Processing** 原始檔轉換到 **Java** 原始檔格式，然後編譯成 **Applet**。同時，系統也會產生一些相關檔案，連同 **Applet** 檔案，儲存在 **sketch** 目錄下的 **applet** 目錄。

產生的相關檔案有

index.html : HTML 檔案，包含 **applet** 及一個 **link**

Sketch_123.jar : Java Archive

Sketch_123.java : Java 原始檔

Sketch_123.pde : Processing 原始檔

loading.gif : 程式下載圖示

在匯出時，系統會將 **data** 目錄下的檔案全部打包起來。因此，產生的 **JAR** 檔案可能會很大，若在 **data** 目錄下，有 **applet** 不會用到的檔案，可以在匯出前先移除，或者在匯出後再利用壓縮軟體將 **JAR** 檔案中不需要的檔案移除，如此就可以將 **JAR** 檔案減小。

請注意：在匯出 **Applet** 時，系統會將先前 **applet** 目錄刪除，然後再寫入新的。此外由於 **Applet** 的安全機制，當在瀏覽器上執行 **Applet** 時，程式是不能存取本機電腦的檔案，也不能連結讀取網站以外主機的檔案，只能讀取網站伺服器的檔案。

1-5. Application 匯出

雖然在 **Processing** 的編輯環境中，可以直接以完整的 **Java** 程式語言來編寫程式，但是只能以匯出 **Applet** 的方式在瀏覽器中執行，如果想要使用一些與存取本機電腦資源有關的程式功能的話，會因 **Applet** 在瀏覽器中執行而有所限制。基於安全性的考量，在瀏覽器中執行 **Applet** 程式，是禁止存取本機電腦資源的，否則就會變成駭客程式了。

因此，需要將 **Applet** 轉換成 **Application**，如此就可以跳脫瀏覽器，單獨地直接在本機電腦上執行程式，也就可以完全地使用所有的功能，同時，也可以以全螢幕的方式來呈現影像。

Application 匯出的步驟：

1. 點選 **File** → **Export Application** 即可。
2. 執行後會在 **sketch** 目錄下產生幾個資料夾，分別是 **application.linux**、**application.macosx**、**application.windows**。
3. **window** 的使用者，執行 **application.windows\ Sketch_123.exe**，即可開啓程式。

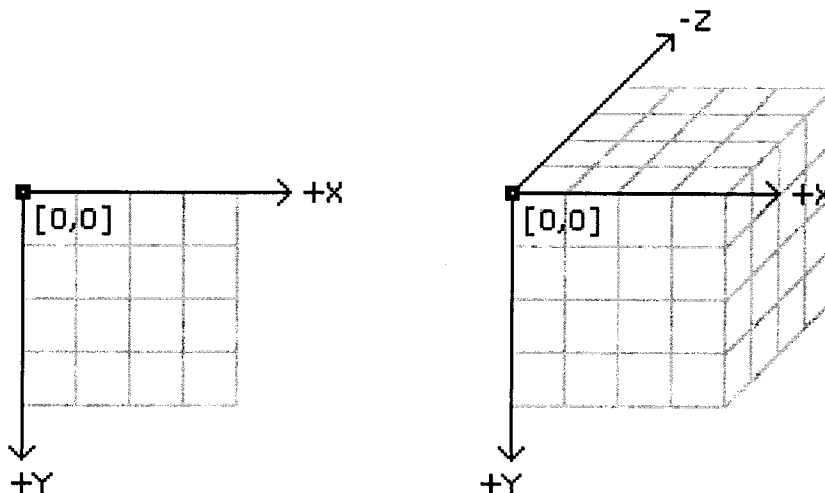
1-6. 全螢幕顯示

1. 在編輯環境中，點選 **Sketch**→**Present** 即可。
2. 在程式中加入以下程式段，也可以在執行 **Sketch_123.exe** 或 **Sketch_123.jar** 時，以全螢幕方式呈現。

```
static public void main(String args[]) {  
    PApplet.main(new String[] { "--present", "YourClassName" });  
}
```

1-7. 影像座標系統

Processing 採用 **Cartesian** 座標系統，原點在左上方，如圖所示，**x** 軸正值向右，**y** 軸正值向下，**z** 軸正值向前，負值向後。而視角預設在螢幕中間，也就是中心點在 $[\text{width}/2, \text{height}/2]$ 。若有一個程式設定圖像大小是 **320 pixels X 240 pixels**，則 $[0,0]$ 代表在左上角， $[320,240]$ 在右下角。



1-8. 影像顏色系統

Processing 的色彩模式分爲 2 種，RGB 模式及 HSB 模式。

預設的模式是 RGB 模式，顏色的設定方式如下所示：

```
// 設定顏色爲 R:204, G:153, B:0, 每個數值範圍爲 0~ 255
color c1 = color(204, 153, 0); // 以 RGB 建立一個顏色數值

// 設定顏色爲 #FFCC00, 以 16 進位的數值表示
color c2 = #FFCC00; // 以 16 進位的數值建立一個顏色數值
noStroke(); // 設定無框線
fill(c1); // 設定填滿顏色
rect(0, 0, 25, 100);
fill(c2);
rect(25, 0, 50, 100);
color c3 = get(10, 50); // 取得 座標(10,50) 的顏色
fill(c3);
rect(50, 0, 75, 100);
```

RGB 模式, 就是 Red 紅, Green 綠, Blue 藍

HSB 模式, 就是 Hue 色相, Saturation 彩度(飽和度), Brightness 亮度

1-9. 主程式架構 (Programming Modes)

Processing 的程式編輯模式有 3 種，基本模式(Basic Mode)、連續模式(Continuous Mode) 以及 Java 進階模式(Java Mode)。初學者最好先從基本模式學起，藉以熟悉座標、變數、迴圈等的操作，然後再轉到連續模式及 Java 進階模式編寫程式。

1. 基本模式

這個模式用來繪製靜態影像及練習程式的基本功能，簡單的畫線的程式碼可

以直接在螢幕上呈現出來，以下這個範例會畫一個黃色的長方形在螢幕上。

```
size(200, 200);    // 設定影像大小 200x200
background(255);  // 設定背景顏色
noStroke();       //設定無框線
fill(255, 204, 0); //設定填入的顏色
rect(30, 20, 50, 50); //從座標[30,20]處, 畫一個 50x50 的長方形.
```

2. 連續模式

這個模式提供一個 `setup()` 及 `draw()` 程式碼區塊，程式一開始會先執行 `setup()` 區塊的程式碼，而且只會執行一次，一般用來設定初始值及環境參數。接著會反覆執行 `draw()` 區塊的程式碼，另外在 `setup()` 及 `draw()` 程式碼區塊之外，可以自行再加入其他的程式碼，像是函式、類別、鍵盤事件和滑鼠事件。

以下範例會畫 4 個圓圈，而畫圓圈的方式是利用呼叫 `circles()` 函式的方式來完成。`circles()` 函式並不是 Processing 本身提供的功能，而是在此程式中另外編寫的。此外，在 `draw()` 區塊中的程式碼只會執行一次，因為在 `setup()` 區塊中，設定了 `noLoop()`，就是不要反覆執行。

```
void setup(){
  size(200, 200);
  noStroke();
  background(255);
  fill(0, 102, 153, 204);
  smooth(); noLoop();
}

void draw(){
  circles(40, 80);
  circles(90, 70);
}

void circles(int x, int y) {
  ellipse(x, y, 50, 50);
  ellipse(x+20, y+20, 60, 60);
}
```

```
}
```

以下這個範例會依據滑鼠的位置來畫長方形，滑鼠的座標儲存在系統變數 (mouseX and mouseY) 中。在 draw() 的區塊中，程式會不斷地執行，用來偵測滑鼠的移動，直到程式被中止。此範例可以產生移動及互動的效果。

```
void setup(){
  size(200, 200);
  rectMode(CENTER);
  noStroke();
  fill(0, 102, 153, 204);
}
void draw(){
  background(255);
  rect(width-mouseX, height-mouseY, 50, 50);
  rect(mouseX, mouseY, 50, 50);
}
```

3. Java 進階模式

這個模式是一個更有彈性的模式，可以在 Processing 編輯環境中，直接以完整的 Java 程式語言來編寫程式。在此模式中，可以使用的程式庫不再受限於 Processing 所提供的程式庫，而是可以使用所有 Java 語言提供的程式庫。

```
public class MyDemo extends PApplet {
  void setup() {
    size(200, 200);
    rectMode(CENTER);
    noStroke();
    fill(0, 102, 153, 204);
  }

  void draw() {
    background(255);
    rect(width-mouseX, height-mouseY, 50, 50);
    rect(mouseX, mouseY, 50, 50);
  }
}
```

```
}
```

1-10. 影像產生模式 (Rendering Modes)

Processing 提供的影像產生模式有以下 3 種，第一是利用 Java 2D 繪圖程式庫，第二是利用一個叫做 P3D 的 3D 繪圖引擎，第三是透過 JOGL 來與 OpenGL 介面溝通。至於要決定使用哪一種模式來輸出影像，可以利用 `size()` 函數來設定。值得一提的是，Processing 花了很多努力，在讓不同的影像產生模式結果相近，但是目前還是有一些差異。

1. JAVA2D

利用 Java 2D 繪圖程式庫來產生 2D 影像。這是預設的影像產生模式，如果沒有特別設定，就會以這個模式產生影像。

```
size(200, 200);  
background(255);  
noStroke();  
fill(255, 204, 0);  
rect(30, 20, 140, 160);
```

以下範例是明確地設定要使用 Java 2D 影像產生模式

```
size(200, 200, JAVA2D);  
background(255);  
noStroke();fill(255, 204, 0);  
rect(30, 20, 140, 160);
```

2. P3D

最適合用來快速繪製 3D 影像，也支援 2D 影像

```
size(200, 200, P3D);  
background(0);  
noStroke();fill(204, 204);  
translate(width/2, height/2);  
rotateX(PI/6);
```

```
rotateY(PI/3);  
rect(-60, -60, 120, 120);
```

3. OPENGL

這個模式適用於有使用支援 OpenGL 圖形加速卡的情況，可以加速圖形的產生，特別適用於要繪製更多圖形以及更大顯示範圍的場合。

Processing 透過 JOGL 來與 OpenGL 介面溝通，其中 JOGL 是由 SUN 的遊戲科技小組所發起的。因此，需要在電腦上安裝支援 OpenGL 的圖形加速卡，才可以充分發揮 JOGL 所提供的程式庫的功能。

採用此模式產生影像，其效果會因所使用的繪圖卡不同及 OpenGL 支援的程度而有所不同。

以下以使用 OPENGL 影像產生模式的範例：

```
import processing.opengl.*;  
  
void setup() {  
  size(800, 600, OPENGL);  
  noStroke();  
}  
  
void draw() {  
  float x = mouseX - width/2;  
  background(255);  
  fill(0, 102, 153, 26);  
  for(int i=0; i<height; i+=10) {  
    for(int j=0; j<width; j+=10) {  
      beginShape(QUADS);  
      vertex(x+j, i, 0);  
      vertex(x+j, i+5, 0);  
      vertex(j-x, i+5, -400);  
      vertex(j-x, i, -400);  
      endShape();  
    }  
  }  
}
```



```
}
```

1-11. 基本程式結構

1. 程式碼及註解 **Statements and Comments**

Statements 是程式碼的基本元素，以分號作為 **Statement** 的結束。

Comments 註解是以雙斜線(//)作為陳述的開始。或是利用 **/** **/** 將要註解的內容包起來。

以下是一個範例

```
// Statements and Comments  
// by REAS <http://reas.com>
```

```
/**
```

```
The size function is a statement that tells the computer  
how large to make the window.
```

```
Each function statement has zero or more parameters.
```

```
Parameters are data passed into the function
```

```
and used as values for specifying what the computer will do.
```

```
**/
```

```
size(200, 200);
```

```
// The background function is a statement that tells the computer
```

```
// which color to make the background of the window
```

```
background(102);
```

2. 座標的使用 **Coordinates**

所有圖形要繪製的時候，都要設定座標來表示位置所在，每個座標值代表與原點之間的距離，以 **pixel** 為單位。其中原點 **[0,0]**是在視窗的左上方，而右下方的座標是 **[width-1, height-1]**，**width** 及 **height** 是視窗的的長和寬。

```

// Coordinates
// by REAS <http://reas.com>

// Sets the screen to be 200, 200, so the width of the window is 200 pixels
// and the height of the window is 200 pixels
size(200, 200);
background(0);
noFill();
stroke(255);

// The two parameters of the point() method each specify coordinates.
// This call to point() draws at the position [100, 100]
point(width/2, height/2); //繪出點

// Draws to the position [100, 50]
point(width/2, height/4);

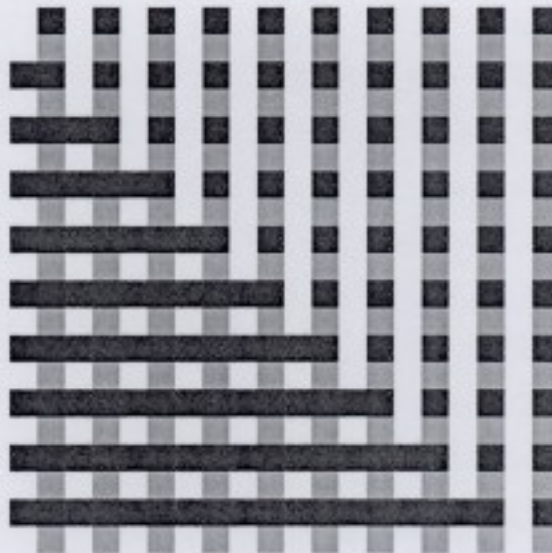
// It is also possible to specify a point with any parameter,
// but only coordinates on the screen are visible
point(60, 30);
point(60, 134);
point(160, 50);
point(280, -800);
point(201, 100);

// Coordinates are used for drawing all shapes, not just points.
// Parameters for different methods are used for different purposes.
// For example, the first two parameters to line() specify the coordinates of
the
// first point and the second two parameters specify the second point
stroke(204); // 設定框線顏色
line(0, 73, width, 73); // 繪出直線

// The first two parameters to rect() are coordinates
// and the second two are the width and height
rect(110, 55, 40, 36); //繪出矩形
ellipse(56, 46, 55, 55); //繪出橢圓形

```

3. 顯示視窗的長和寬 Width and Height



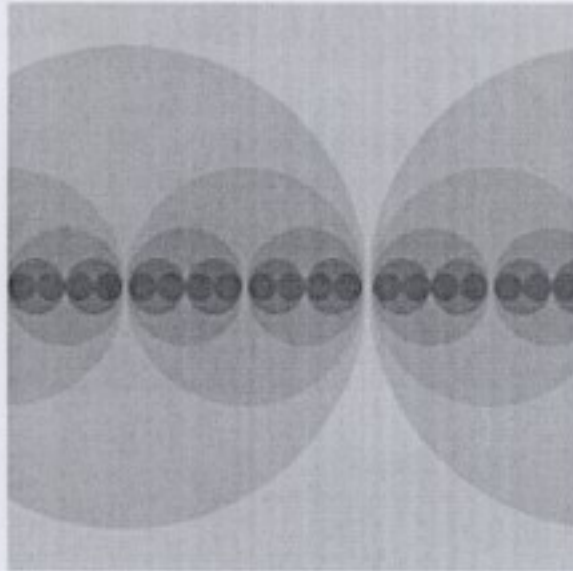
當利用 `size()` 函數設定顯示視窗的長與寬後，就可以利用 `'width'` 及 `'height'` 這 2 個變數來取得長與寬的數值了。

```
// Width and Height
// by REAS <http://reas.com>

size(200, 200);
background(127);
noStroke();
for(int i=0; i<height; i+=20) {
  fill(0);
  rect(0, i, width, 10);
  fill(255);
  rect(i, 0, 10, height);
}
```

1-12. 進階程式結構

1. 遞迴 Recursion (自我呼叫函式)



以下的範例是利用自我呼叫函式的方式來達到遞迴的效果，不過終止點的設定要很小心。否則，若產生無限遞迴的情況，很容易使程式當掉。

以下範例的終止點，是變數 **level** 等於 1 時就停止遞迴。

```
// Recursion
// by REAS <http://reas.com>

void setup() {
  size(200, 200);
  noStroke();
  smooth();
  noLoop();
}

void draw() {
  drawCircle(126, 170, 6);
}

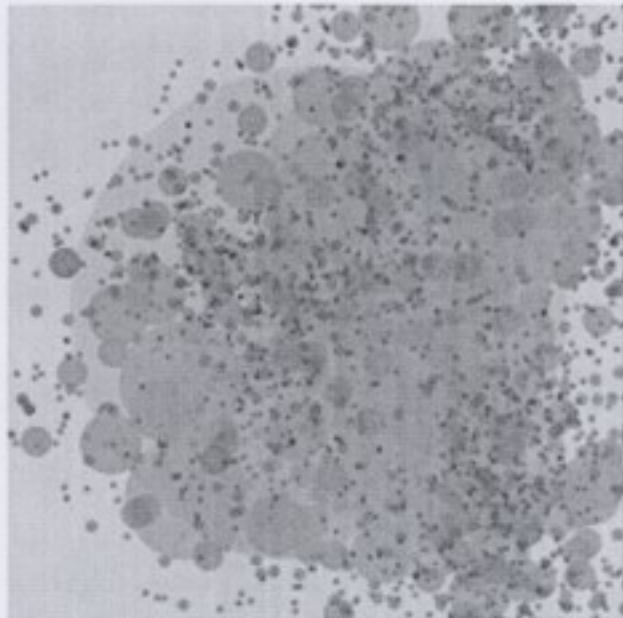
void drawCircle(int x, int radius, int level) {
```

```

float tt = 126 * level/4.0;
fill(tt);
ellipse(x, 100, radius*2, radius*2);
if(level > 1) {
  level = level - 1;
  drawCircle(x - radius/2, radius/2, level);
  drawCircle(x + radius/2, radius/2, level);
}
}

```

以下是另一個遞迴的範例



```

// Recursion
// by REAS <http://reas.com>

void setup() {
  size(200, 200);
  noStroke();
  smooth();
  drawCircle(100, 100, 80, 8);
}

void drawCircle(float x, float y, int radius, int level) {

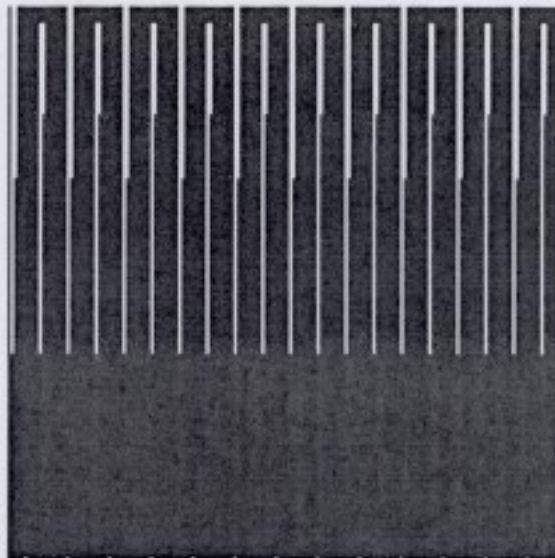
```

```

float tt = 126 * level/6.0;
fill(tt, 153);
ellipse(x, y, radius*2, radius*2);
if(level > 1) {
  level = level - 1;
  int num = int(random(2, 6));
  for(int i=0; i<num; i++) {
    float a = random(0, TWO_PI);
    float nx = x + cos(a) * 6.0 * level;
    float ny = y + sin(a) * 6.0 * level;
    drawCircle(nx, ny, radius/2, level);
  }
}
}
}

```

2. 使用物件 Objects



此範例是當滑鼠滑過影像時，幾何圖形的位置及速度會改變。在程式中，定義一個由一組線條所構成的物件，然後利用呼叫物件的函數，就可以使每一個物件產生各別的動作及反應。

```

// Objects
// By hbarragan

MRect r1, r2, r3, r4;

```

```

void setup() {
    size(200, 200);
    fill(255);
    noStroke();
    r1 = new MRect(1, 134.0, 0.532, 0.083*height, 10.0, 60.0);
    r2 = new MRect(2, 44.0, 0.166, 0.332*height, 5.0, 50.0);
    r3 = new MRect(2, 58.0, 0.332, 0.4482*height, 10.0, 35.0);
    r4 = new MRect(1, 120.0, 0.0498, 0.913*height, 15.0, 60.0);
}

```

```

void draw() {
    background(0);

    r1.display();
    r2.display();
    r3.display();
    r4.display();

    r1.moveToX(mouseX-(width/2), 30);
    r2.moveToX((mouseX+(width*0.05))%width, 20);
    r3.moveToX(mouseX/4, 40);
    r4.moveToX(mouseX-(width/2), 50);

    r1.moveToY(mouseY+(height*0.1), 30);
    r2.moveToY(mouseY+(height*0.025), 20);
    r3.moveToY(mouseY-(height*0.025), 40);
    r4.moveToY((height-mouseY), 50);
}

```

```

class MRect {
    int w; // single bar width
    float xpos; // rect xposition
    float h; // rect height
    float ypos ; // rect yposition
    float d; // single bar distance
    float t; // number of bars
    float side; // rect width
}

```

```

MRect(int iw, float ixp, float ih, float iyp, float id, float it) {
    w = iw;
    xpos = ixp;
    h = ih;
    ypos = iyp;
    d = id;
    t = it;
}

void moveToY (float posY, float damping) {
    float dif = ypos - posY;
    if (abs(dif) > 1) {
        ypos -= dif/damping;
    }
}

void moveToX (float posX, float damping) {
    float dif = xpos - posX;
    if (abs(dif) > 1) {
        xpos -= dif/damping;
    }
}

void display() {
    for (int i = 0; i<t; i++) {
        rect(xpos+(i*d*w), ypos, w, height*h);
    }
}
}

```


1-13. 程式庫 (Processing API)

http://processing.org/reference/index_ext.html

Processing 提供許多程式庫, 指令, 讓藝術家們可以很方便將創作的想法實現.

目前提供的程式庫分為以下幾個大類別, 由於數量很多, 請自行參考線上的說明. 另外會針對不同的主題, 說明程式庫的使用方式.

Structure 程式結構

Environment 程式執行環境

Data 資料

Data – Primitive 基本型別

Data – Composite 複合型別

Data – Conversion 型別轉換

Data - String Functions 字串處理

Data - Array Functions 矩陣處理

Control 程序控制

Control - Relational Operators 比較關係運算

Control – Iteration 迴圈

Control – Conditionals 條件

Control - Logical Operators 條件關係運算

Shape 圖形

Shape - 2D Primitives 2D 基本形

Shape – Curves 曲線

Shape - 3D Primitives 3D 基本形

Shape – Attributes 圖形屬性

Shape – Vertex 圖形端點

Input 輸入

Input – Mouse 滑鼠

Input – Keyboard 鍵盤

Input – Files 檔案

Input – Web 網站

Input - Time & Date 時間, 日期

Output 輸出

Output - Text Area 訊息文字

Output -Image 儲存影像

Output -Files 儲存檔案

Transform 3D 轉換

Lights, Camera 光源, 視角

Lights, Camera - Lights 光源

Lights, Camera - Camera 視角

Lights, Camera - Coordinates 座標

Lights, Camera - Material Properties 材質

Color 顏色

Color - Setting 設定

Color - Creating & Reading 建立, 讀取

Image 影像

Image - Loading & Displaying 載入, 顯示

Image - Pixels 圖素

Rendering 圖形呈現

Typography 文字

Typography - Loading & Displaying 載入, 顯示

Typography - Attributes 屬性

Typography - Metrics 對齊

Math 數學

Math - Operators 基本運算

Math - Bitwise Operators 位元運算

Math - Calculation 計算

Math - Trigonometry 三角函數

Math - Random 隨機函數

Constants 常數

1-14. 延伸閱讀

1. 範例程式

<http://processing.org/learning/index.html>

請參考線上範例程式，藉由多看程式碼，了解程式邏輯與作品呈現之間的關係，當有看不懂的地方，可以查看線上的程式指令說明，同時也可以從所提供的說明，了解該指令的使用方式，並且複製該程式段到自己的程式之中。

目前線上的範例程式，包括有：Basics、Topics、3D & OpenGL、Libraries 等。

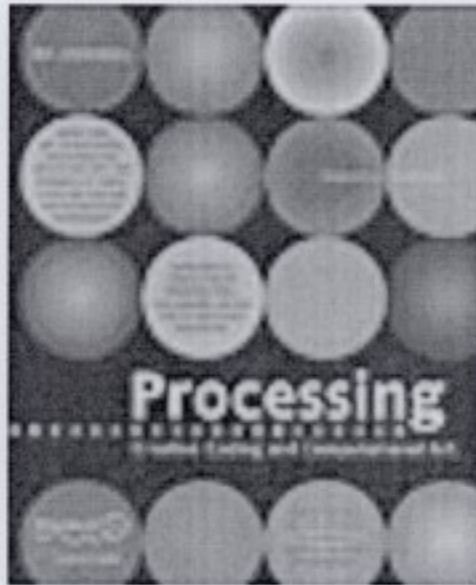
2. Processing 書籍

另外目前已有幾位作者出版 Processing 相關的書籍，例如：

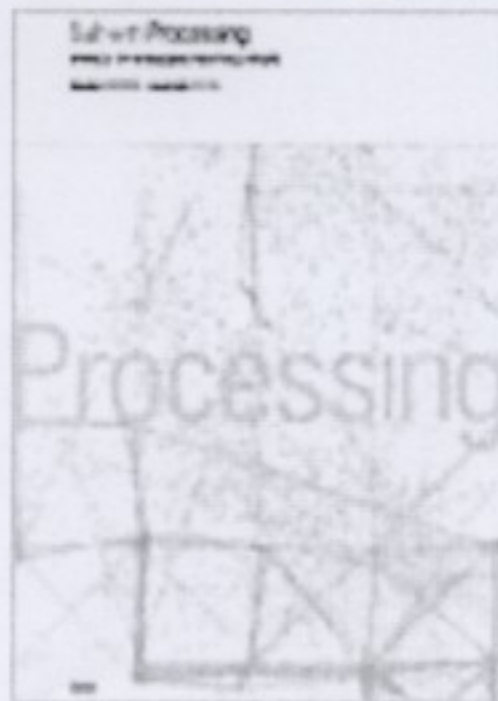


Processing: A Programming Handbook for Visual Designers and Artists.

Casey Reas and Ben Fry. Published 24 August 2007 by MIT Press. 736 pages.



Processing: Creative Coding and Computational Art.
Ira Greenberg. Published 28 May 2007 by Friends of Ed. 840 pages.



Built with Processing (Japanese).
Published 28 March 2007 by BNN.

第二章、Processing 程式與軟體安裝

【Webcam 與多媒體創作】工作坊

大綱：

2-1. 所需軟體

2-2. 所需硬體

2-3. 自備器材

2-4. 安裝步驟

2-1. 所需軟體

1. Basic

* Java SDK

(JDK 6u2), <http://java.sun.com/javase/downloads/index.jsp>

(jdk-6u2-windows-i586-p.exe)

(JDK 5.0 Update 12),

http://java.sun.com/javase/downloads/index_jdk5.jsp

(jdk-1_5_0_12-windows-i586-p.exe)

* Processing (0125),

<http://www.processing.org/download/processing-0125-expert.zip>

(processing-0125-expert.zip)

* Demo Program, (Sound *1, Webcam *2, Wii *1)

2. For Microphone Sound

* Sound Lib : Minim(1.1),

<http://code.compartmental.net/minim/distro/minim-1.1-lib.zip>

(minim-1.1-lib.zip)

3. For Webcam Vision

*Vision Lib : BlobDetection,

<http://www.v3ga.net/processing/BlobDetection/Resources/9/>

(blobDetection.jar)

*QuickTime (7.2), <http://www.apple.com/quicktime/download/>

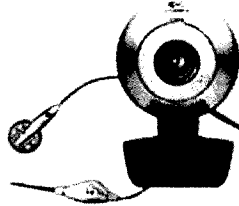
(QuickTimeInstaller.exe)

*WinVDIG, (104) http://www.vdig.com/WinVDIG/WinVDIG_104.exe

(WinVDIG_101.exe, WinVDIG_104.exe)

*Webcam Driver : Logitech,

http://www.logitech.com/index.cfm/support_downloads/downloads/&cl=tw,z
[h](http://www.logitech.com/index.cfm/support_downloads/downloads/&cl=tw,z)



QuickCam® Messenger (qc1100.exe)

QuickCam® IM™ (qc1100.exe)



QuickCam® for Notebooks Pro (qc848cht.exe)

4. For Wii Remote

*Wii Lib :

WiiRemoteJ(1.0)

<http://www.world-of-cha0s.hostrocket.com/WiiRemoteJ/> (WiiRemoteJ v1.0.zip.gz)

*BlueTooth Lib :

avetanaBluetooth

<http://www.avetana-gmbh.de/avetana-gmbh/produkte/bestellung.eng.xml>

*BlueTooth Driver :

MSI, (Star Key 2.0 (3X Faster)) (WIDCOMM Bluetooth Dongle Driver)

http://taiwan.msi.com.tw/index.php?func=downloaddetail&type=driver&maincat_no=131&prod_no=101 (StarKey_4.0.1.2900.zip)



2-2. 所需硬體

1. Webcam+Microphone :

Logitech 羅技 快看傳訊家 QuickCam Messenger (內建麥克風)

http://shopping.pchome.com.tw/?mod=item&func=exhibit&IT_NO=ABAL18-A03124781&SR_NO=ABAN1P&ROWNO=6

2. BlueTooth Adapter 藍芽接收器 :

MSI 微星 Star Key 2.0

http://taiwan.msi.com.tw/index.php?func=downloaddetail&type=driver&maincat_no=131&prod_no=101

3. Wii Remote :

Wii 遙控器 <http://buy.yahoo.com.tw/gdsale/gdsale.asp?gdid=489988>

2-3. 自備器材

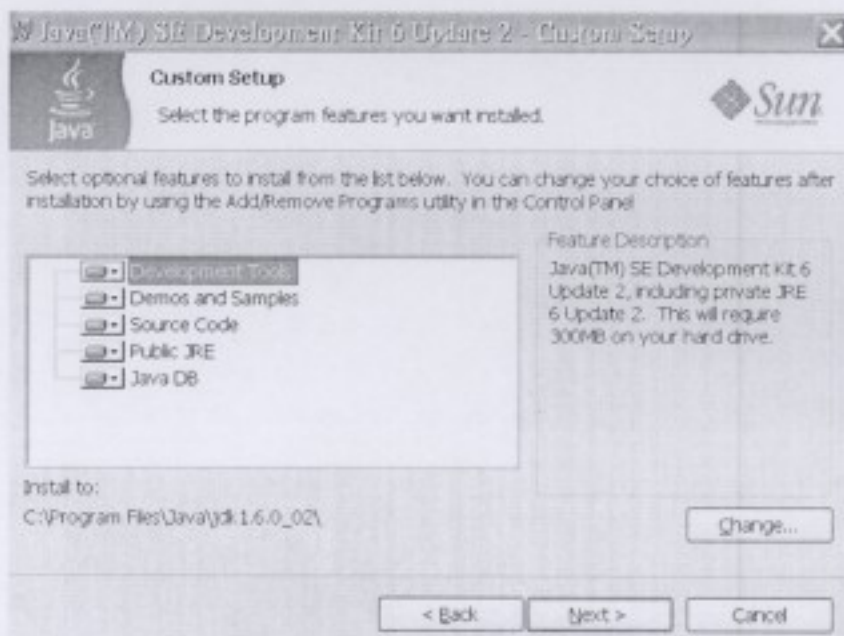
1. 筆記型電腦或個人電腦
2. Webcam+Microphone (羅技的 Webcam 較穩定)
3. BlueTooth Adapter 藍芽接收器 (Star Key 2.0)
(ps.務必要使用微星的藍芽接收器 (Star Key 2.0), 不要用筆電內建的, 因為經測試只有微星的藍芽接收才可以與 Processing 整合。)
4. Wii Remote 遙控器

2-4. 安裝步驟

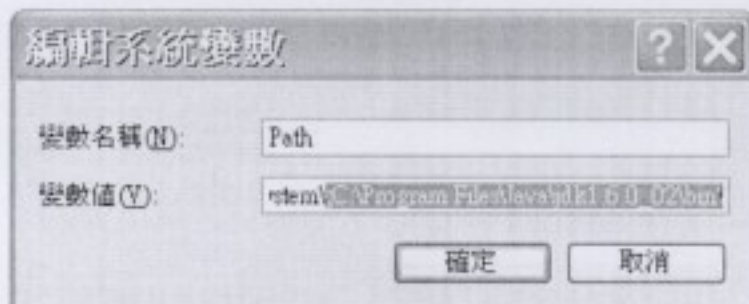
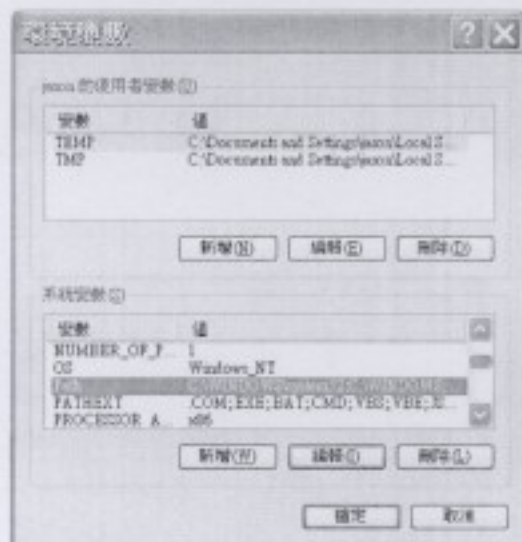
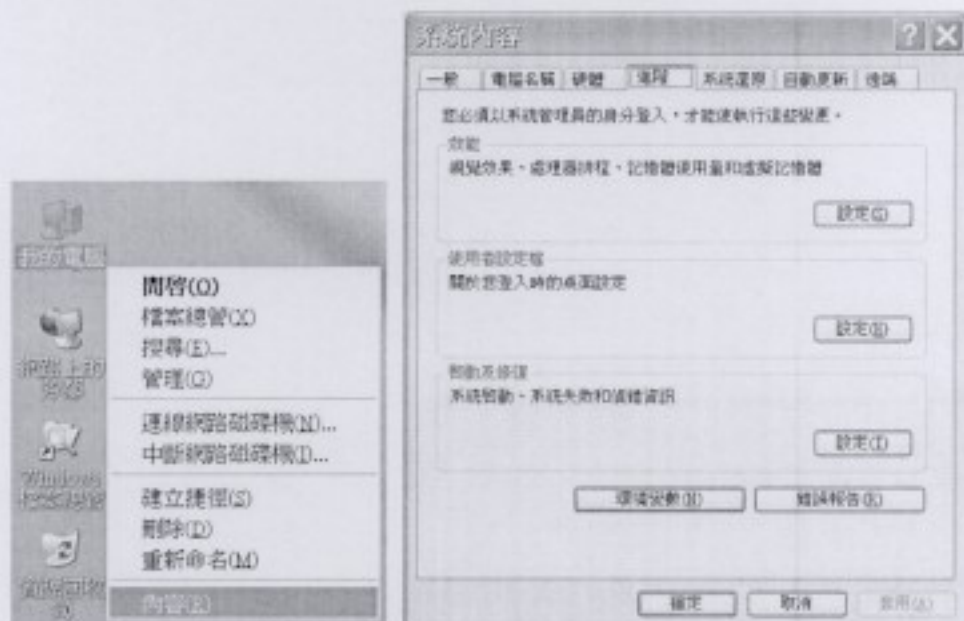
**Basic **

1. Java SDK (JDK 6u2)

■ 執行 `jdk-6u2-windows-i586-p.exe`.

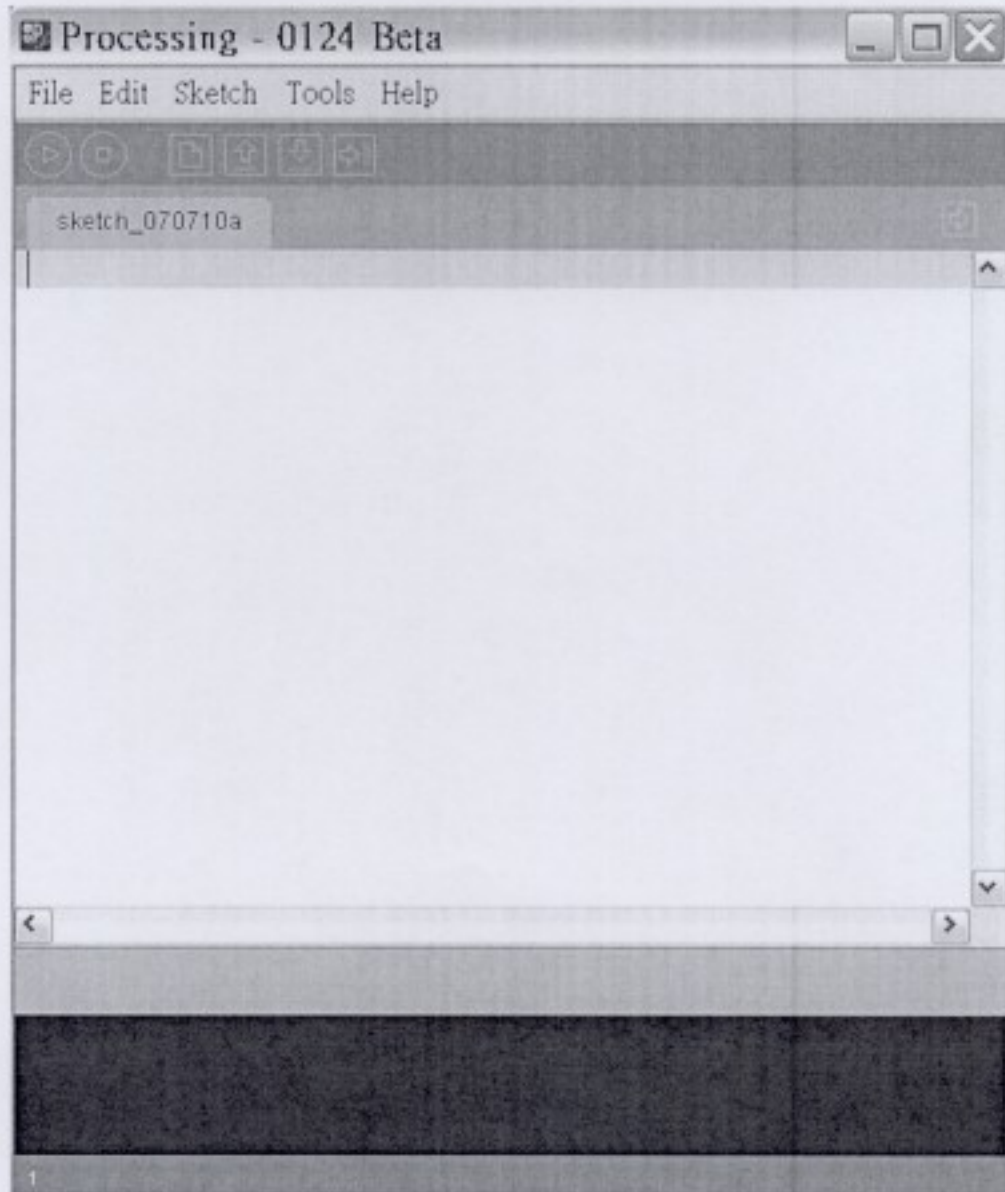


■ 設定 path 路徑：[我的電腦]→[內容]→[進階]→[環境變數]→[系統變數]，點選 Path，按下[編輯]，在變數值內容的最後處 加入 ;C:\Program Files\Java\jdk1.6.0_02\bin\ 即可。

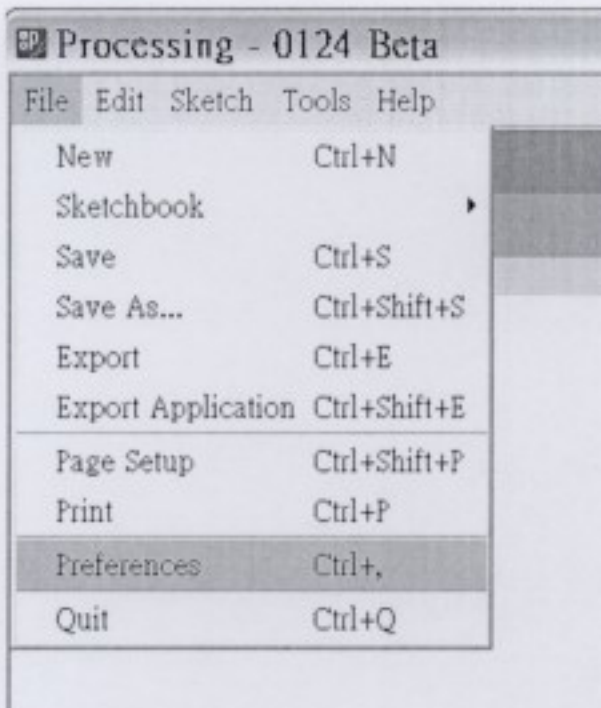


2. Processing (0125)

- 請將 processing-0125-expert.zip 解壓縮至 C:\ 解壓縮後會在 C:\目錄下出現 processing-0125-expert 資料夾。
- 執行 C:\ processing-0125-expert\ processing.exe, 若可以順利開啓程式就表示安裝成功。



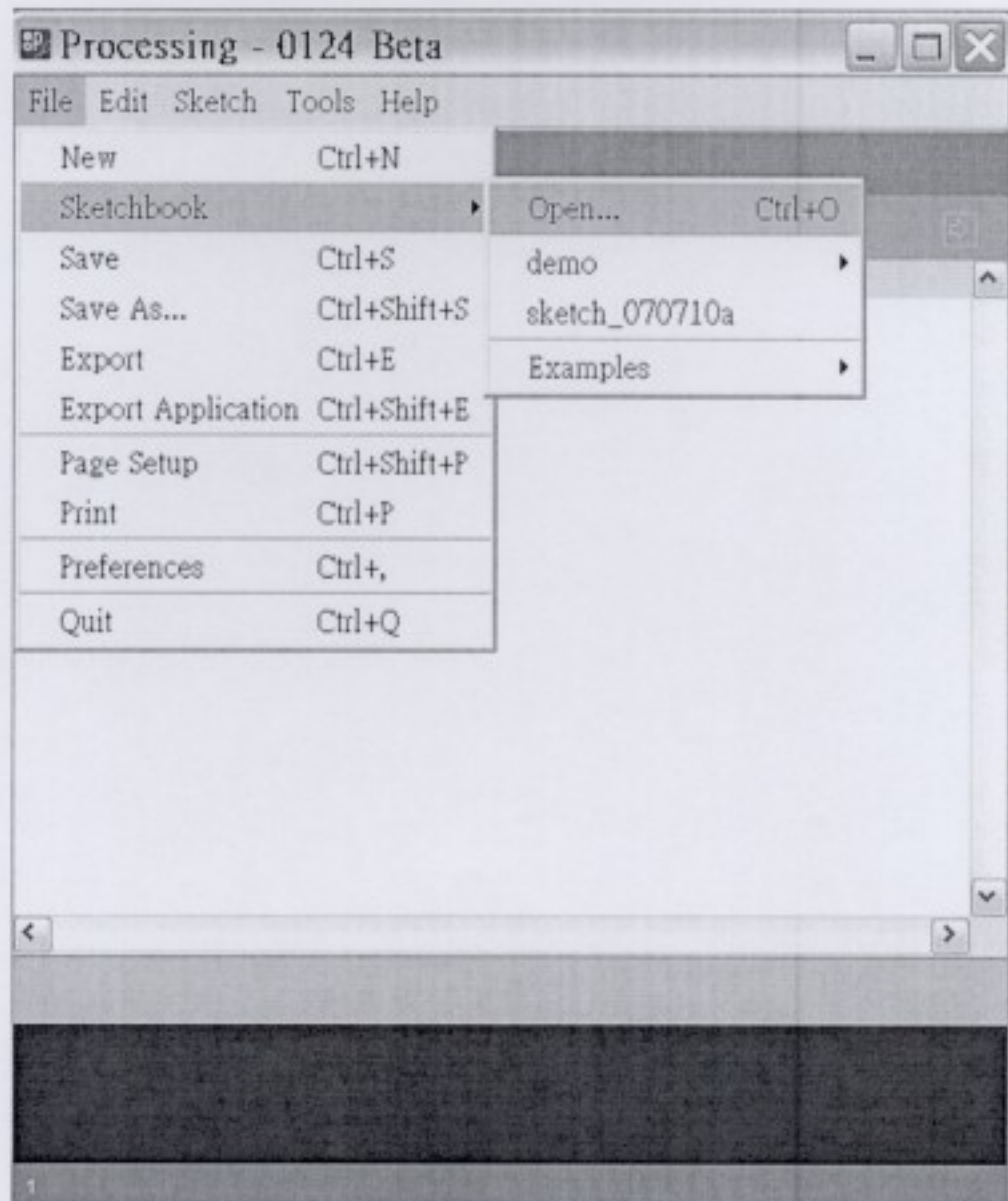
- 設定工作目錄, 請在 C:\ 目錄下新增一個資料夾 artwork,
- 開啓 processing, 點選 File-> Preferences 選項, 在 Preferences 對話框中, 設定 Sketchbook location 為 C:\ artwork.



3. Demo Program

■ 請將 demo.zip 解壓縮至 C:\artwork，解壓縮後會在 C:\artwork 目錄下出現 demo 資料夾

■ 開啓 processing 後，在 File-> Sketchbook 選項下，會看到 demo 及 Examples 項目。



**** For Webcam Vision *****

1. Vision Lib : BlobDetection

■ 請將 blobDetection 資料夾複製到 C:\ processing-0125-expert\libraries 目錄下。

2. Webcam Driver : Logitech,

■ 安裝 webcam 驅動程式

■ 請先不要插入 webcam, 直到安裝程式提示時再插入

■ QuickCam® Messenger (qc1100.exe)

■ QuickCam® IM™ (qc1100.exe)

■ QuickCam® for Notebooks Pro (qc848cht.exe)

3. QuickTime (7.2),

■ 安裝 QuickTimeInstaller.exe

4. WinVDIG, (101)

■ 安裝 WinVDIG_101.exe

■ 完成後執行開始->程式集->WinVDIG->QTCap，若可以看到 webcam 影像， 就表示安裝成功。

■ 執行示範程式

**** For Microphone Sound *****

Sound Lib : Minim(1.1)

■ 請將 minim 資料夾複製到 C:\ processing-0125-expert\libraries 目錄下

■ 執行示範程式

**** For Wii Remote ****

1. Wii Lib : WiiRemoteJ(1.0)

■ 請將 WiiRemoteJ 資料夾複製到 C:\ processing-0125-expert\libraries 目錄下

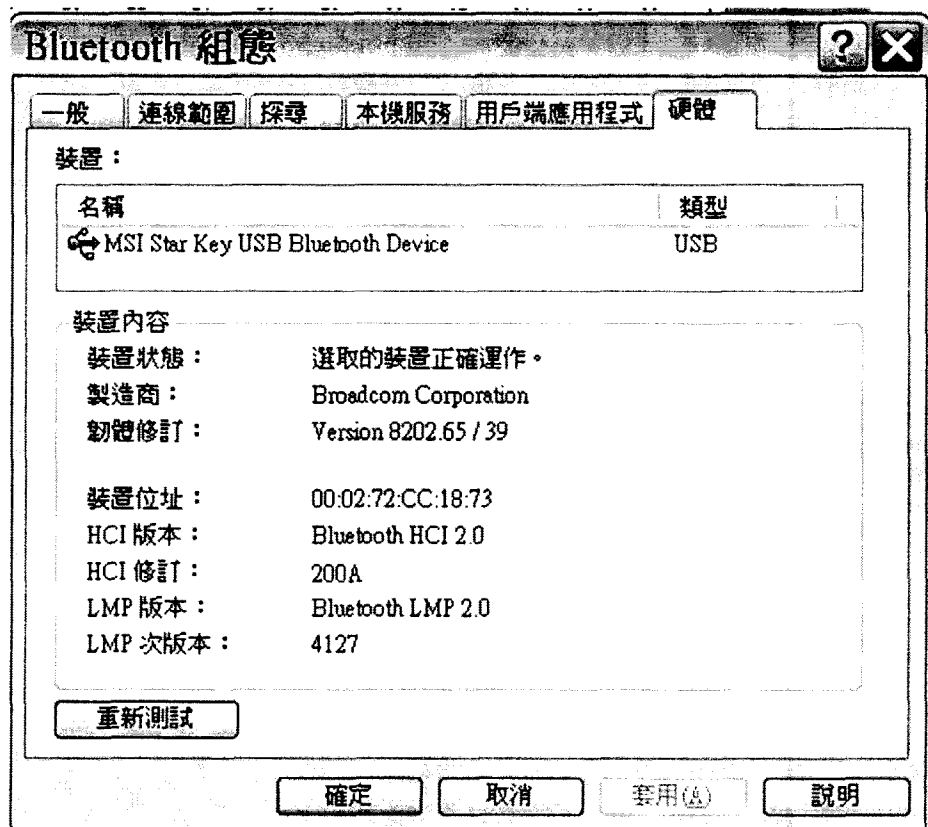
2. BlueTooth Driver : MSI, (Star Key 2.0 (3X Faster)) (WIDCOMM Bluetooth Dongle Driver))

■ 請先不要插入藍芽接收器，直到安裝程式提示時再插入。

■ 安裝 Star Key 驅動程式，執行 StarKey_4.0.1.2900 目錄下之 setup.exe

■ 安裝後，請在右下角藍芽圖示上，按滑鼠右鍵，選擇「進階組態設定」，在「硬體」的項目下，檢看是否出現藍芽接收器裝置，並抄下裝置位置，例如： 00:02:72:cc:18:73





■ 若出現找不到的情況，請將藍芽接收器拔出再插入，並按下「重新測試」即可。

■ BlueTooth Lib : avetanaBluetooth, 請連至

<http://www.avetana-gmbh.de/avetana-gmbh/produkte/bestellung.eng.xml>

■ 輸入基本資料, email, 藍芽接收器裝置位置等資料後後, 按 submit 送出, 請注意 BT-Address 的格式是 00-02-72-cc-18-73

Name	jason	
Firstname	lee	
Company	aaa	
Street	taipei	
PO Box City	100	
Phone	0800000123	
Your download-link will be sent to this address. Please do not enter a dummy address.		
E-Mail	jasonlee2357@giga.net.tw	
Notify me of updates in avetanaBluetooth	<input checked="" type="checkbox"/>	
Please make sure to enter at least one valid Bluetooth-Address, that you want to use this library on (e.g. 00-0d-93-05-17-0e)		
BT-Address 1	00-02-72-cc-18-73	
BT-Address 2		
BT-Address 3		
<input type="button" value="Submit"/>		

■ 之後查看一下自己的 email, 是否有收到一封指引下載點的信, 打開信後, 連至該網頁, 點選「this link.」, 下載程式庫 avetanaBluetooth.jar, 並儲存至 C:\processing-0125-expert\libraries\WiiRemoteJ\library . 目錄下

To download your library, please click (Save Link target as...) [this link.](#)

The license will be valid until 23.07.2007.
The licens is right now valid for the following Bluetooth Addresses:

You must enter at least one Bluetooth-Address for which the license shall be valid (e.g. 00-0d-93-05-17-0e)

Bluetooth-Address 1	<input type="text" value="00-02-72-cc-18-73"/>
Bluetooth-Address 2	<input type="text" value="00-02-72-cc-18-73"/>
Bluetooth-Address 3	<input type="text" value="00-02-72-cc-18-73"/>

To change the Addresses for which the library shall be valid, please enter the new addresses in this form and submit the product has not been purchased.

■ 執行示範程式, 在執行程式時要先進行藍芽裝置的連線, 當藍芽接收器出現閃爍藍燈時, 按下 **Wii Remote** 後方的紅色按鈕, 就會開始進行連線, 等到 **Wii Remote** 的 4 個閃爍藍燈變成一個藍燈時, 就是連線成功。基本上當程式結束時會執行斷線的動作, 若因一些原因導致無法執行斷線的動作, 再按一下 **Wii Remote** 後方的紅色按鈕, 就可以完成斷線了。這個部份很重要, 因為若沒有執行斷線的動作, 下一次再執行程式時, 就會找不到 **Wii Remote**。

第三章、Processing 程式指令與範例說明

【Webcam 與多媒體創作】工作坊

大綱

3-1. Processing 基本指令

3-2. Processing 範例說明

本次的程式示範，主要以 **processing** 所提供的程式為主，因為數量很多，學員可以將程式碼複製到編輯器中，修改其中的參數數值，函數等，以了解數值的改變對於呈現的影響，即可將不同的修改結果另存新檔。

3-1. Processing 基本指令

1. Color 顏色

```
background(0);    // 設定背景顏色
background(255);
background(255, 204, 0);
stroke(255);      //設定線條顏色
stroke(0);
stroke(255, 204, 0);
fill(0, 102, 153); //設定填滿顏色
```

2. Shape 形狀

```
point(30, 20);    // 畫點
line(0, 20, 80, 20); // 畫線, 由(0,20) 畫到 (80,20)
rect(10, 20, 30, 40); // 畫矩形, 在(10,20)處, 畫一個寬 30 高 40 的矩形
ellipse(56, 46, 55, 80); // 畫橢圓, 在(56,46)處, 畫一個寬直徑 55 高直徑 80 的圓形
```

3. Image 影像

```
set(30, 20, 255); // 設定影像座標(30,20)的顏色為 R:30, G:20, B:255  
a = get(60, 10); // 取得影像座標(60,10)的顏色，並指定給 a 變數。
```

```
pixels[y*width+x]
```

// 取得影像座標(x, y)的顏色，由於一個尺寸 width x height 的影像資料，在 processing 中是以在 pixels 的一維陣列中，所以若要取知道影像座標(x, y)在 pixels 陣列中的位置，利用 y*width+x 來得到，而 pixels[y*width+x]就是影像座標(x, y)的顏色

4. 資料

```
int x = 70; // Initialize //宣告一個整數變數 x, 並設定整數變數 x 的值為 70  
x = 30; // Change value // 設定整數變數 x 的值為 30  
float x = 70.0; // 宣告一個浮點數變數 x, 並設定浮點數變數 x 的值為 70.0  
x = 30.0; // 設定浮點數變數 x 的值為 30.0  
int[] a = {5, 10, 11}; // 建立一個陣列 a, 並設定陣列元素為 5, 10, 11。  
a[0] = 12; // Reassign // 設定陣列 a 的第 0 個元素為 12。
```

5. Control 控制

無限次數反覆執行

```
void draw() {  
    // Statements 反覆執行此程式段之程式碼  
}
```

有限次數反覆執行

```
for(int a=45; a<=55; a++) {  
    // Statements 反覆執行此程式段之程式碼, 由 a=45 執行到 a=55,  
    每執行一次 a 就加 1,  
    // a++ 代表 a=a+1;  
}
```

有條件執行

```
if(c==1) {  
    // Statements, 當 c 等於 1 時, 執行此程式段之程式碼  
}  
  
if(c!=1) {  
    // Statements, 當 c 不等於 1 時, 執行此程式段之程式碼  
}  
if(c < 1) {  
    // Statements, 當 c 小於 1 時, 執行此程式段之程式碼  
}  
if(c >= 1) {  
    // Statements, 當 c 大於或等於 1 時, 執行此程式段之程式碼  
}  
  
if((c >= 1) && (c < 20)) {  
    // Statements, 當 c 大於或等於 1 以及 c 小於 20 時, 執行此程式段  
    之程式碼  
}  
  
if(c >= 20) {  
    // Statements 1, 當 c 大於或等於 20 時, 執行此程式段之程式碼  
} else if (c == 0) {  
    // Statements 2, 否則當 c 等於 0 時, 執行此程式段之程式碼  
} else {  
    // Statements 3, 否則執行此程式段之程式碼  
}
```

6. Input/Output 輸入/輸出

mouseX 取得滑鼠 x 座標
mouseY 取得滑鼠 y 座標
mousePressed 取得滑鼠是否按下之狀態

```
void mousePressed() {  
    // Statements 當滑鼠按下時, 執行此程式段之程式碼  
}
```

(key=='a') 取得鍵盤按下之數值，並判斷是否為字元 a
(key=='b') 取得鍵盤按下之數值，並判斷是否為字元 b

...

```
void keyPressed() {  
    // Statements 當鍵盤按下時，執行此程式段之程式碼  
}
```

hour() 取得目前時間之小時數值

minute() 取得目前時間之分鐘數值

second() 取得目前時間之秒數數值

3-2. Processing 範例說明

1. Statements and Comments

Statements are the elements that make up programs. The ";" (semi-colon) symbol is used to end statements. It is called the "statement terminator." Comments are used for making notes to help people better understand programs. A comment begins with two forward slashes ("//").

Statements 陳述式，程式碼是構成程式的基本元素，";" (semi-colon) 分號是用來作為一段陳述式或程式碼的結束符號。 **Comments** 註解 是用來對於程式碼外加的說明，有助於理解程式碼的作用，slashes ("//") 雙斜線符號表示此符號之後的文字為註解說明文字。

```
// The size function is a statement that tells the computer  
// how large to make the window.  
// Each function statement has zero or more parameters.  
// Parameters are data passed into the function  
// and used as values for specifying what the computer will do.
```

```
size(200, 200);
```

```
// The background function is a statement that tells the computer
```

```
// which color to make the background of the window
```

```
background(102);
```

2. Coordinates

All shapes drawn to the screen have a position that is specified as a coordinate. All coordinates are measured as the distance from the origin in units of pixels. The origin [0, 0] is the coordinate in the upper left of the window and the coordinate in the lower right is [width-1, height-1].

所有圖形都是利用座標的設定來繪製，可以設定圖形出現的位置，圖形的大小，長度，寬，及高，圖形的單位為圖素(pixel)，整張繪製區域的座標為，左上角(0,0) 右下角(width-1, height-1)

```
// Sets the screen to be 200, 200, so the width of the window is 200 pixels
```

```
// and the height of the window is 200 pixels
```

```
size(200, 200); // 設定畫布大小
```

```
background(0); // 設定背景顏色
```

```
noFill(); // 設定無填滿
```

```
stroke(255); // 設定線條顏色
```

```
// The two parameters of the point() method each specify coordinates.
```

```
// This call to point() draws at the position [100, 100]
```

```
point(width/2, height/2); // 畫點
```

```
// Draws to the position [100, 50]
```

```
point(width/2, height/4); // 畫點
```

```
// It is also possible to specify a point with any parameter,
```

```
// but only coordinates on the screen are visible
```

```
point(60, 30); // 畫點
```

```
point(60, 134);
```

```

point(160, 50);
point(280, -800);
point(201, 100);

// Coordinates are used for drawing all shapes, not just points.
// Parameters for different methods are used for different purposes.
// For example, the first two parameters to line() specify the coordinates
of the
// first point and the second two parameters specify the second point
stroke(204); //設定線條顏色
line(0, 73, width, 73); // 畫線

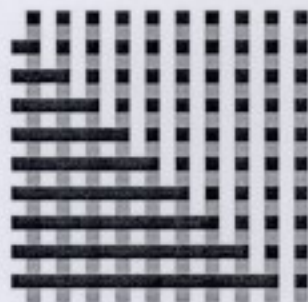
// The first two parameters to rect() are coordinates
// and the second two are the width and height
rect(110, 55, 40, 36); // 畫矩形

```

3. Width and Height

The 'width' and 'height' variables contain the width and height of the display window as defined in the size() function.

寬(width) 及 高(height)，代表展示視窗的寬跟高，利用 size(200, 200) 來設定視窗的寬跟高。



```

size(200, 200);
background(127);
noStroke();
for(int i=0; i<height; i+=20) {
  fill(0);
  rect(0, i, width, 10);
}

```

```
fill(255);
rect(i, 0, 10, height);
}
```

4. Setup and Draw

The `draw()` function creates a structure in which to write programs that change with time.

在 `setup()` 的程式段中，只會執行一次，一般用來作初始狀態的設定。在 `draw()` 的程式段中，會反覆執行下去。

```
// The statements in the setup() function
// execute once when the program begins
void setup() {
  size(200, 200); // Size should be the first statement
  stroke(255);    // Set line drawing color to white
  frameRate(30); // 設定 draw()程式段，執行的速率。
}
```

```
float y = 100;
```

```
// The statements in draw() are executed until the
// program is stopped. Each statement is executed in
// sequence and after the last line is read, the first
// line is executed again.
void draw() {
  background(0); // Set the background to black
  y = y - 1;
  if (y < 0) { y = height; }
  line(0, y, width, y);
}
```

5. No Loop

The `noLoop()` function causes `draw()` to only execute once. Without calling `noLoop()`, `draw()` executed continually.

`noLoop()` 指令會使 `draw()` 程式段只執行一次，或者使 `draw()` 程式段不再反覆執行下去。

```

// The statements in the setup() function
// execute once when the program begins
void setup() {
  size(200, 200); // Size should be the first statement
  stroke(255);    // Set line drawing color to white
  frameRate(30);
  noLoop();
}

```

```
float y = 100;
```

```

// The statements in draw() are executed until the
// program is stopped. Each statement is executed in
// sequence and after the last line is read, the first
// line is executed again.

```

```

void draw() {
  background(0); // Set the background to black
  y = y - 1;
  if (y < 0) { y = height; }
  line(0, y, width, y);
}

```

6. Loop

The loop() function causes draw() to execute continuously. If noLoop is called in setup() the draw() is only executed once. In this example click the mouse to execute loop(), which will cause the draw() the execute continuously.

loop()指令會使 draw()程式段開始反覆執行下去。

```

// The statements in the setup() function
// execute once when the program begins
void setup() {
  size(200, 200); // Size should be the first statement
  stroke(255);    // Set stroke color to white
  noLoop();
}

```



```

float y = 100;

// The statements in draw() are run until the
// program is stopped. Each statement is run in
// sequence and after the last line is read, the first
// line is run again.
void draw() {
    background(0); // Set the background to black
    line(0, y, width, y);

    y = y - 1;
    if (y < 0) {
        y = height;
    }
}

void mousePressed() {
    loop(); // 當滑鼠按下時，執行 loop()指令，使得 draw()程式段開始反覆執行下去。
}

```

7. Re-draw

The `redraw()` function makes `draw()` execute once. In this example, `draw()` is executed once every time the mouse is clicked. `redraw()`指令會使 `draw()`程式段再執行一次。

```

// The statements in the setup() function
// execute once when the program begins
void setup() {
    size(200, 200); // Size should be the first statement
    stroke(255); // Set line drawing color to white
    noLoop();
}

float y = 100;

// The statements in draw() are executed until the

```

```

// program is stopped. Each statement is executed in
// sequence and after the last line is read, the first
// line is executed again.
void draw() {
  background(0); // Set the background to black
  y = y - 1;
  if (y < 0) { y = height; }
  line(0, y, width, y);
}

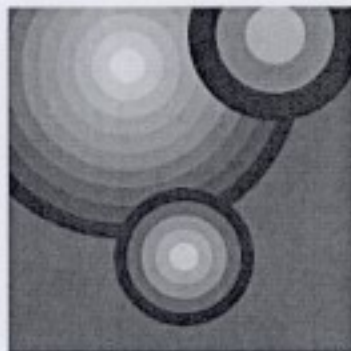
void mousePressed() {
  redraw(); // 當滑鼠按下時，執行 redraw ()指令，使得 draw()程式段再執行一次。
}

```

8. Functions

The `draw_target()` function makes it easy to draw many distinct targets. Each call to `draw_target()` specifies the position, size, and number of rings for each target.

Function 是指自訂指令函數，可以將經常反覆會使用到的程式段，放入自訂的指令函數中，再特別注意的是輸入到此自訂指令的參數有哪些？從自訂指令輸出的傳回值是什麼。在範例中，自訂指令 `draw_target()` 用來畫出一個同心圓的圖形，其輸入參數是 `int xloc` (圓心 x 座標), `int yloc` (圓心 y 座標), `int size` (同心圓直徑), `int num` (同心圓圈數), `void` 代表沒有傳回值，



```

void setup() {
  size(200, 200);
  background(51);
  noStroke();
}

```

```

    smooth();
    noLoop();
}

void draw() {
    draw_target(68, 34, 200, 10); // 在座標(68,34)處繪出一個直徑 200，圈數為 10 的
    同心圓。
    draw_target(152, 16, 100, 3);
    draw_target(100, 144, 80, 5);
}

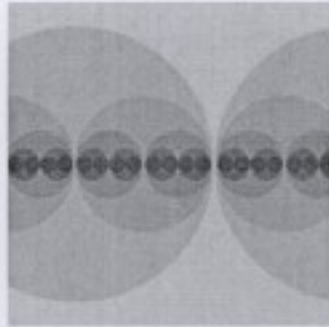
void draw_target(int xloc, int yloc, int size, int num) {
    float grayvalues = 255/num; // 依據同心圓圈數，計算每一同心圓的灰階值的差
    距。
    float steps = size/num; // 依據同心圓圈數，計算每一同心圓的直徑的差距
    for(int i=0; i<num; i++) {
        fill(i*grayvalues); // 設定圓形的填滿的灰階值
        ellipse(xloc, yloc, size-i*steps, size-i*steps); // 在座標(xloc, yloc)處畫出直徑
        size-i*steps 的圓形
    }
}

```

9. Recursion

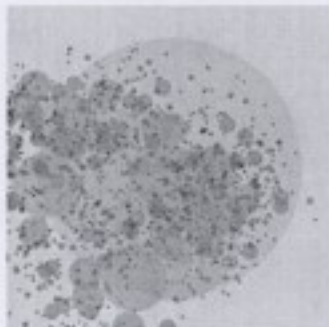
A demonstration of recursion, which means functions call themselves. Notice how the drawCircle() function calls itself at the end of its block. It continues to do this until the variable "level" is equal to 1.

以下範例說明 自訂指令函數 呼叫 本身指令函數為形成一種遞迴迴圈的情形，使用此方法時，要注意迴圈停止點的條件設定，如此才不造成無窮遞迴的情形。範例中，從初始的直徑開始畫圓，畫的當中又再以半徑的一半，繼續再畫圓下去，依此類推，直到 level 值小於等於 1 時才停止畫圓。此時出現的畫面是一種不斷縮小的呈現效果。自訂指令 drawCircle() 用來畫出圓形，其輸入參數是 int x(圓心座標), int radius(圓形半徑), int level (遞迴階數)



```
void setup() {  
  size(200, 200);  
  noStroke();  
  smooth();  
  noLoop();  
}  
  
void draw() {  
  drawCircle(126, 170, 6); // 在座標 x 為 126 處，畫出半徑 170 的圓，  
  階數為 6。  
}  
  
void drawCircle(int x, int radius, int level) {  
  float tt = 126 * level/4.0; // 計算填滿圓形的灰階值  
  fill(tt);  
  ellipse(x, 100, radius*2, radius*2); // 在座標(x, 100) 畫出直徑為  
  radius*2 的圓。  
  if(level > 1) {  
    level = level - 1;  
    drawCircle(x - radius/2, radius/2, level);  
    drawCircle(x + radius/2, radius/2, level);  
  }  
}
```

另一個遞迴圓圈範例



```

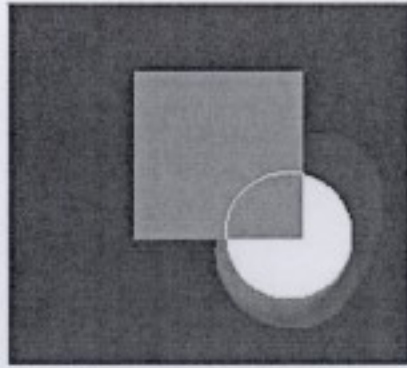
void setup() {
  size(200, 200);
  noStroke();
  smooth();
  drawCircle(100, 100, 80, 8);
}

void drawCircle(float x, float y, int radius, int level) {
  float tt = 126 * level/6.0;
  fill(tt, 153);
  ellipse(x, y, radius*2, radius*2);
  if(level > 1) {
    level = level - 1;
    int num = int(random(2, 6)); // 在 2 跟 6 之間，取得一個隨機數字
    for(int i=0; i<num; i++) {
      float a = random(0, TWO_PI); // 在 0 跟 TWO_PI 之間，取得一個隨機數字
      float nx = x + cos(a) * 6.0 * level; // 計算新的 nx 座標
      float ny = y + sin(a) * 6.0 * level; // 計算新的 n 座標
      drawCircle(nx, ny, radius/2, level); // 畫出下一層的圓形
    }
  }
}

```

10. Create Graphics

The `createGraphics()` function creates an object from the `PGraphics` class (`PGraphics` is the main graphics and rendering context for Processing). The `beginDraw()` method is necessary to prepare for drawing and `endDraw()` is necessary to finish. Use this class if you need to draw into an off-screen graphics buffer or to maintain two contexts with different properties.



利用 createGraphics() 指令，建立一個 PGraphics 物件，

```
PGraphics pg;

void setup() {
  size(200, 200);
  pg = createGraphics(80, 80, P3D); // 建立一個圖形物件 pg
}

void draw() {
  fill(0, 12); // 加入這 2 行
  rect(0, 0, width, height); // 會產生殘影效果
  fill(255);
  noStroke();
  ellipse(mouseX, mouseY, 60, 60); // 畫出一個實心圓

  pg.beginDraw(); // 開始設定在 PGraphics 上的畫圖形
  pg.background(102);
  pg.noFill();
  pg.stroke(255);
  pg.ellipse(mouseX-60, mouseY-60, 60, 60); // 畫出一個空心圓
  pg.endDraw();

  image(pg, 60, 60); // 將 pg 呈現出來。
}
```

第四章、Java ME 程式與說明

【 Mobile 與多媒體創作 】 工作坊

Java ME 簡介

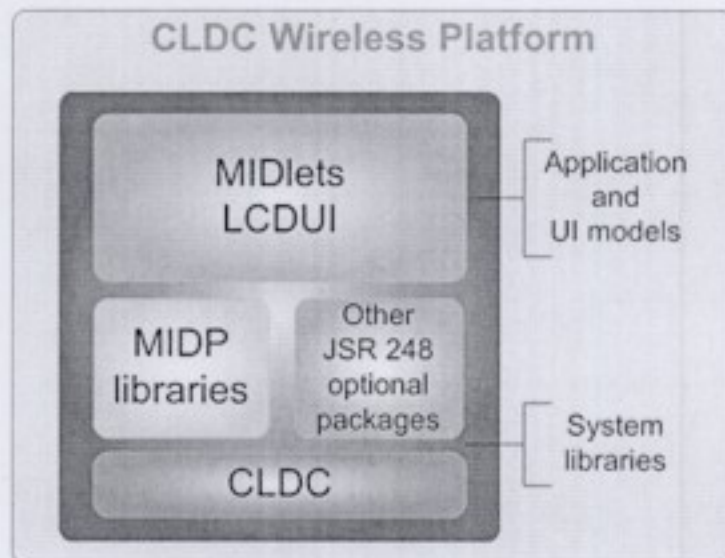
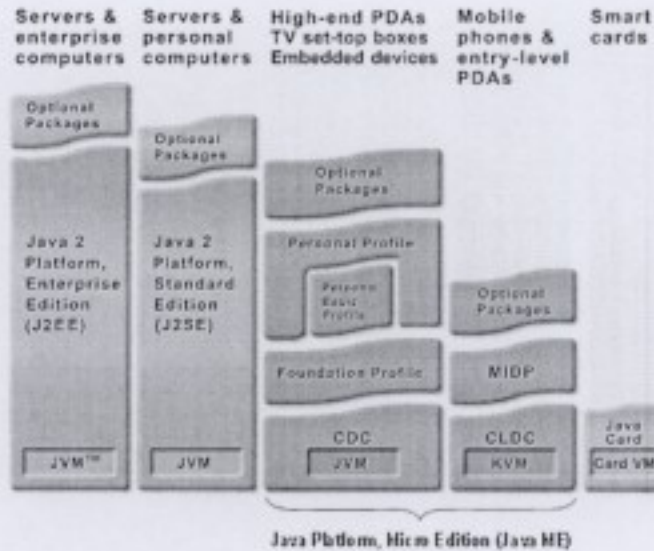
4-1. Java ME Platform CLDC Specification

Java Platform, Micro Edition (Java ME) provides a robust, flexible environment for applications running on mobile and other embedded devices—mobile phones, personal digital assistants (PDAs), TV set-top boxes, and printers.

Java ME includes flexible user interfaces, robust security, built-in network protocols, and support for networked and offline applications that can be downloaded dynamically.

Applications based on Java ME are portable across many devices, yet leverage each device's native capabilities.

<http://java.sun.com/javame/index.jsp>



4-2. Sun Java Wireless Toolkit for CLDC

The Sun Java Wireless Toolkit (WTK) is state-of-the-art toolbox for developing wireless applications that are based on the Java ME platform Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), and designed to run on cell phones, mainstream personal digital assistants, and other small mobile devices.

The toolkit includes the emulation environments, performance optimization and tuning features, documentation, and examples that developers need to bring efficient and successful wireless applications to market quickly.

Sun Java Wireless Toolkit 透過標準的 API 可實作優異的功能。透過 Java Community Process (JCP) 定義 API：Java Specification Request (JSR)

- Mobile Service Architecture (JSR 248)
- Java Technology for the Wireless Industry (JTWI) (JSR 185)
- Connected Limited Device Configuration (CLDC) 1.1 (JSR 139)
- Mobile Information Device Profile (MIDP) 2.0 (JSR 118)
- PDA Optional Packages for the J2ME Platform (JSR 75)
- Java APIs for Bluetooth (JSR 82)
- Mobile Media API (MMAPI) (JSR 135)
- J2ME Web Services Specification (JSR 172)
- Security and Trust Services API for J2ME (JSR 177)
- Location API for J2ME (JSR 179)
- SIP API for J2ME (JSR 180)
- Mobile 3D Graphics API for J2ME (JSR 184)
- Wireless Messaging API (WMA) 2.0 (JSR 205)
- Content Handler API (JSR 211)
- Scalable 2D Vector Graphics API for J2ME (JSR 226)
- Payment API (JSR 229)
- Advanced Multimedia Supplements (JSR 234)
- Mobile Internationalization API (JSR 238)
- Java Binding for the OpenGL(R) ES API (JSR 239)

4-3. Sun Product Offering for Constrained Devices-based on CLDC Configuration

<http://java.sun.com/javame/overview/products.jsp>

For small devices Sun offers a full implementation of an optimized virtual machine based on CLDC together with a set of optional packages. This solution is represented by 2 related and closely connected products:

the CLDC HotSpot Implementation (CLDC.HI) which is a product that is compliant with the Java ME Platform CLDC specification and built for maximum performance within a resource constrained platform.

the Sun Java Wireless Client - which is a pre-integrated and optimized

solution that enables customers to deploy mobile Java technologies with quick time-to-market and reduced cost.

This product is pre-integrated and pre-tested with CLDC HI

The product can be bought either as a Standards Pack which includes all Technology Compatibility Kits for all the included JSRs or it can be licensed as a Wireless Performance Pack License.

In the Performance Pack the Sun optimized implementation of CLDC and Sun Java Wireless Client are included as described above.

The Performance Pack License also includes future updates to the Sun Java Wireless Client, i.e. as new JSRs gets added to the client this will automatically be made available for the customer to use.



Performance Pack License

Sun Java Wireless Client 1.1.3
(plus future upgrades and updates)

JSR 75 File & PIM	JSR 135 Mobile Media	JSR 205 Messaging
JSR 82 Bluetooth	JSR 184 3D Graphics	JSR 226 Vector Graphics
JSR 177 Security & Trust	JSR 118 MIDP	JSR 172 Web Services

CLDC 1.1.3 HotSpot Implementation

Cutting-edge performance	Reduces porting efforts
Minimal footprint	Preserve battery life
Fast application start-up time	

JSRs in orange blocks represent JSRs licensed by companies other than Sun.*

The next version of the Sun Java Wireless Client will include an extended set of JSRs and APIs. The picture below represents the JSRs included in the current version of the Sun Java Wireless Client.

Standards Pack License

TCKs for MSA (JSR 243)

Included component JSRs

JSR 118
MIDP

JSR 135
Mobile Media

JSR 172
Web Services

JSR 177
Security & Trust

JSR 205
Messaging

JSR 211
Content Handler

JSR 75
File & PM

JSR 82
Bluetooth

JSR 179
Location

JSR 180
SIP

JSR 184
3D Graphics

JSR 226
Vector Graphics

JSR 229
Payment

JSR 234
Multimedia Supp

JSR 238
Internationalization

CLDC 1.1
Technology Compatibility Kit (TCK)

JSRs in orange blocks represent JSRs licensed by companies other than Sun.*

Performance Pack License

Sun Java Wireless Client 1.1.4

(plus future upgrades and updates)

JSR 118
MIDP

JSR 135
Mobile Media

JSR 172
Web Services

JSR 177
Security & Trust

JSR 205
Messaging

JSR 211
Content Handler

JSR 75
File & PM

JSR 82
Bluetooth

JSR 179
Location

JSR 180
SIP

JSR 184
3D Graphics

JSR 226
Vector Graphics

JSR 229
Payment

JSR 234
Multimedia Supp

JSR 238
Internationalization

CLDC 1.1.4 HotSpot Implementation

Cutting-edge performance Reduces porting efforts
Minimal footprint Preserve battery life
Fast application start-up time

JSRs in orange blocks represent JSRs licensed by companies other than Sun.*

4-4. Connected Limited Device Configuration (CLDC) HotSpot Implementation

The Connected Limited Device Configuration (CLDC) HotSpot Implementation is Sun's high-performance Java virtual machine for resource-constrained wireless phones and communicator-type devices.

The CLDC HotSpot Implementation is the recommended virtual machine technology for new product deployments in this class of devices, and can be integrated with the Sun Java Wireless Client for a full stack solution using Java technology.

The goal of optimized implementations are to provide a high-performance, high-quality, and flexible implementation that reduces cost and time to market when porting Java ME technologies to targeted handsets.

High Performance - essential for responsive, interactive user experiences in applications written for the Java programming language

High Quality Code - Emphasis on extensive internal testing and documentation and enhances quality in terms of code design and portability

Flexibility - Necessary to support different devices with different processors, operating systems, and native hardware services

4-5. Sun Java Wireless Client

The Sun Java Wireless Client is an optimized implementation that focuses on performance, footprint, portability, and extensibility.

The value propositions of the Sun Java Wireless Client are:

1. Simplified integration for MSA Subset (JSR 248) with well defined native porting interfaces
2. Reduced cost by allowing faster ports, and providing reusable system modules

3. Optimized performance for complete stack and footprint by selective compilation
4. Flexible to customize by providing prebuilt and replaceable modules and build infrastructure
5. Customizable phone environment through the use of adaptive user interface technology
6. Pre-tested with Java Device Test Suite

Sun Java Wireless Client version 1.1.3 today covers the following technologies:

- Mobile Information Device Profile 2.0 (JSR 118)
- Mobile Media APIs (JSR 135)
- Connected Limited Device Configuration 1.1 (JSR 139)
- Wireless Messaging APIs (JSR 205)
- PIM & File System (JSR 75)
- Bluetooth (JSR 82)
- XML Parser (JSR 172)
- Security & Trust (JSR 177)
- 3D Graphics (JSR 184)
- Scalable Vector Graphics (JSR 226)

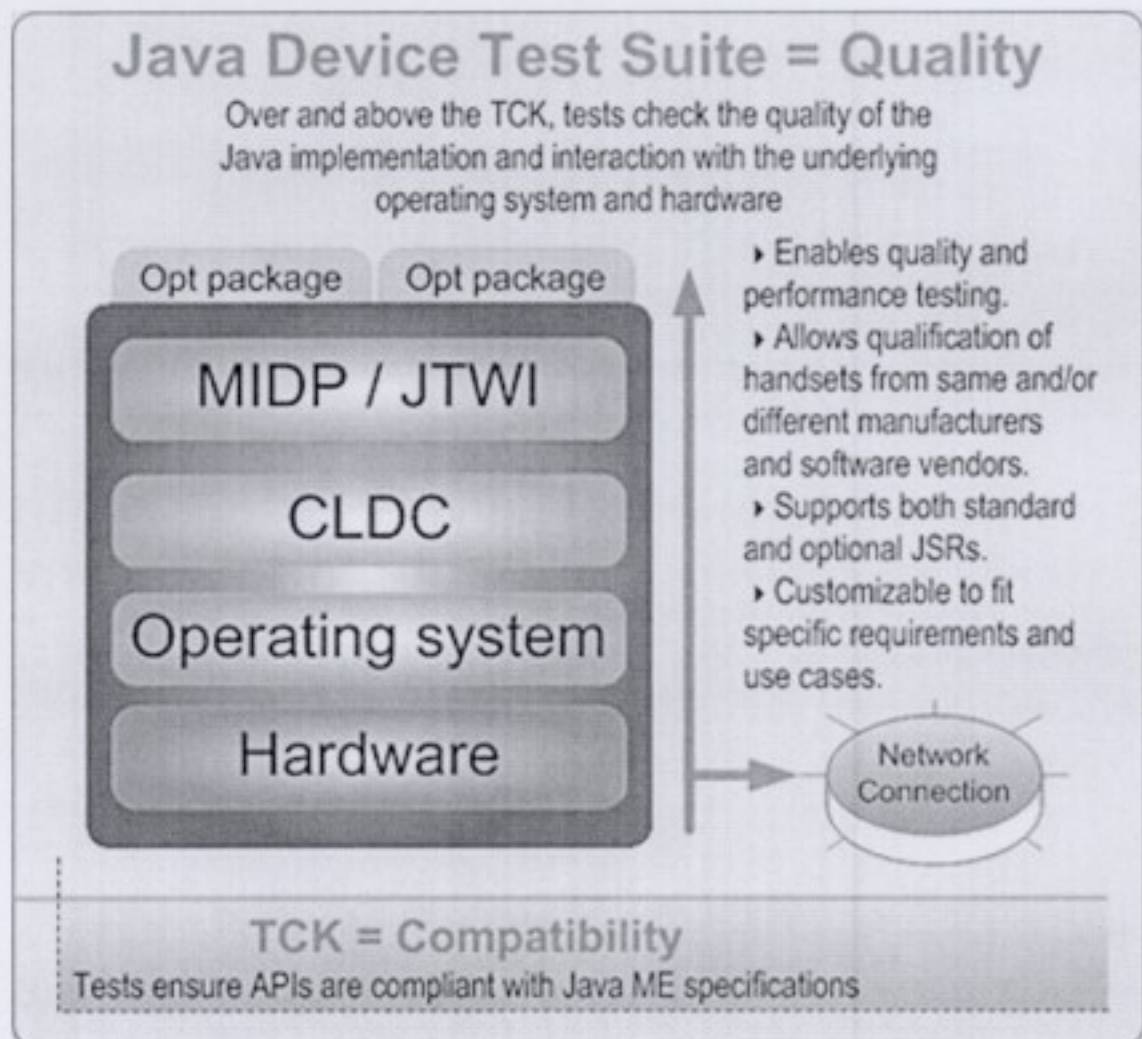
Following this version of the Sun Java Wireless Client the next (version 1.1.4) will support the following additional optional JSRs:

- Location API for J2ME (JSR 179)
- SIP API for J2ME (JSR 180)
- Content Handler API (JSR 211)
- Payment API (JSR 229)
- Advanced Multimedia Supplements (JSR 234)
- Mobile Internationalization API (JSR 238)

4-6. Java Device Test Suite (JDTS)

The Java Device Test Suite (JDTS) simplifies quality assurance and reduces time-to-market for implementations of the Java ME Platform, by providing comprehensive tests and robust test manager to evaluate, validate, and verify the quality of implementations of the Connected Limited Device Configuration (CLDC) and the Mobile Information Device Profile (MIDP) on a particular device.

The main components of the Java Device Test Suite is shown in the picture below:

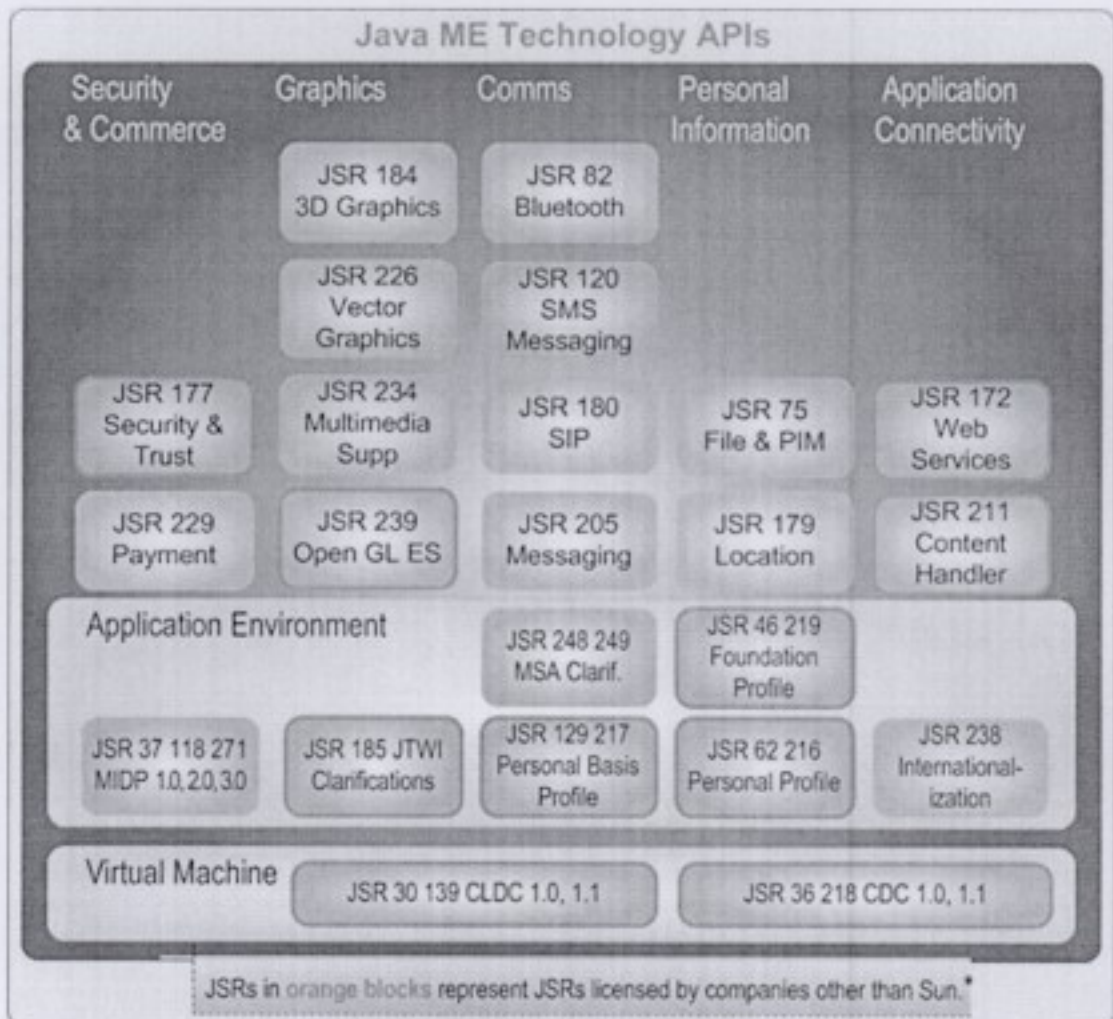


Specific Java ME Technology API offerings

The results from the JCP process for each of the JSRs is as described under Java Community Process a specification, a Reference Implementation (RI) and a Compatibility Test Kit (CTK).

The reference implementation is not delivered as a commercial product, therefore most of the code running on devices today are products from different companies which is based on the specifications.

The picture below represents a summary of the relevant JSRs for the Java ME Platform. For each JSR where Sun has a Reference Implementation and a TCK the JSR is highlighted with a red frame.



Documentation for the Java Platform, Micro Edition (Java ME)

<http://java.sun.com/javame/reference/apis.jsp>

第五章、Mobile Processing 軟體安裝

【 Mobile 與多媒體創作 】 工作坊

Mobile Processing 軟體安裝

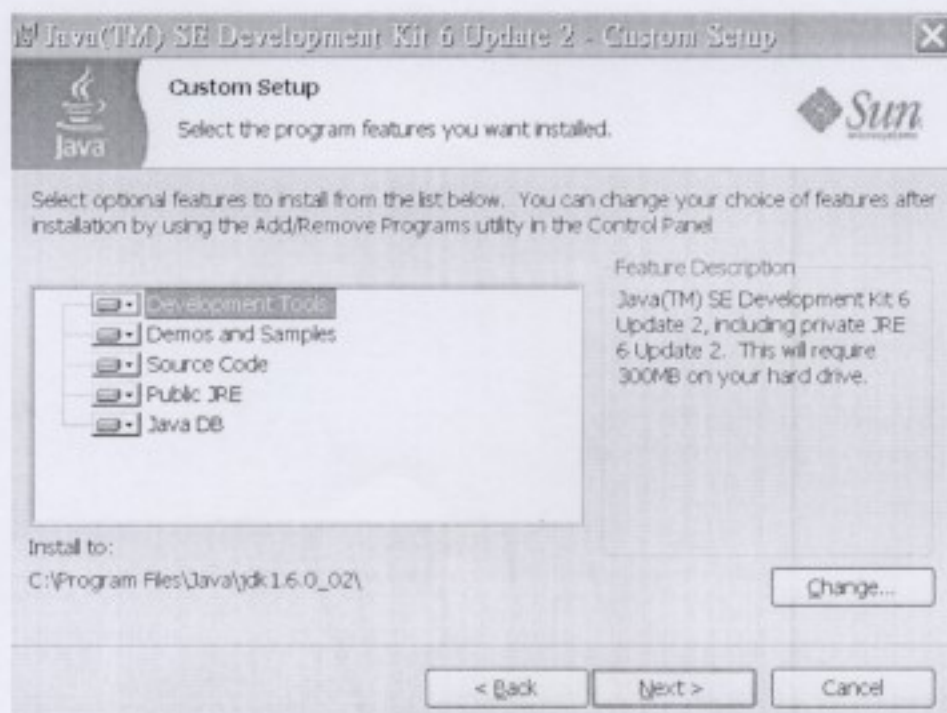
5-1. JDK 6 Update 2

1. 下載網址:

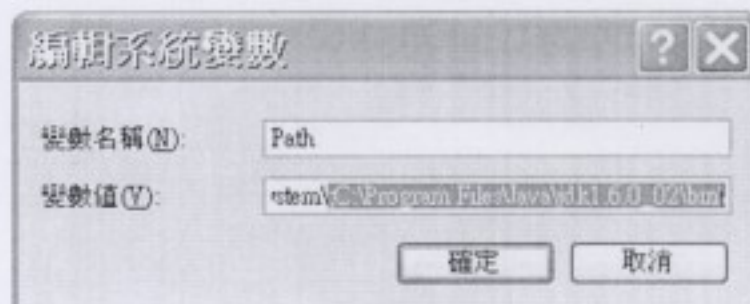
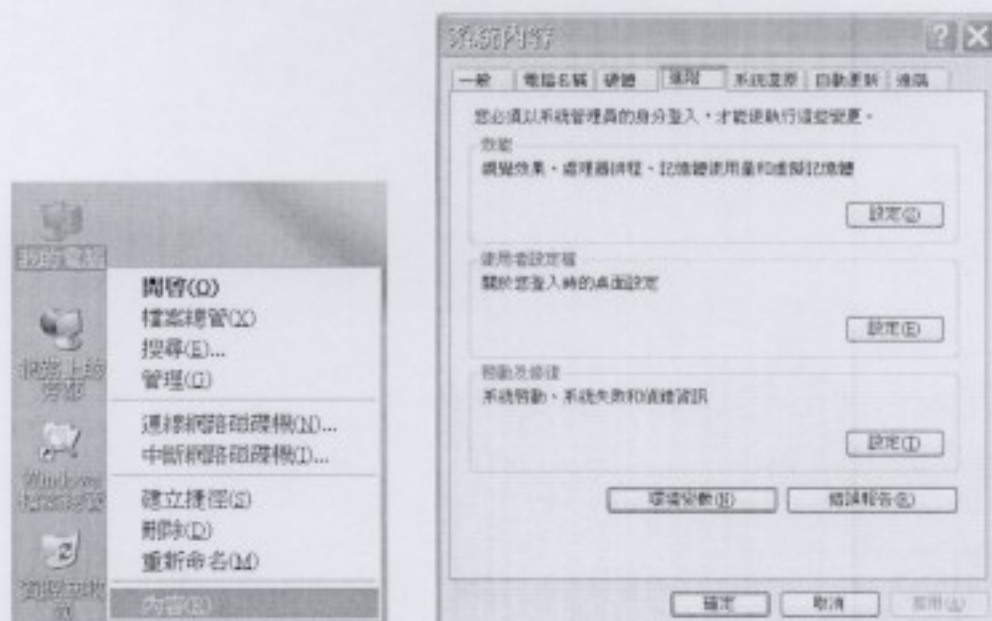
<http://java.sun.com/javase/downloads/index.jsp>

■The Java SE Development Kit (JDK) includes the Java Runtime Environment (JRE) and command-line development tools that are useful for developing applets and applications.

2. 執行 `jdk-6u2-windows-i586-p.exe`.



3. 設定 path 路徑：[我的電腦]→[內容]→[進階]→[環境變數]→[系統變數]，點選 Path，按下[編輯]，在變數值內容的最後處 加入 ;C:\Program Files\Java\jdk1.6.0_02\bin\ 即可。



5-2. Sun Java Wireless Toolkit for CLDC 2.5.2

1. 下載網址:

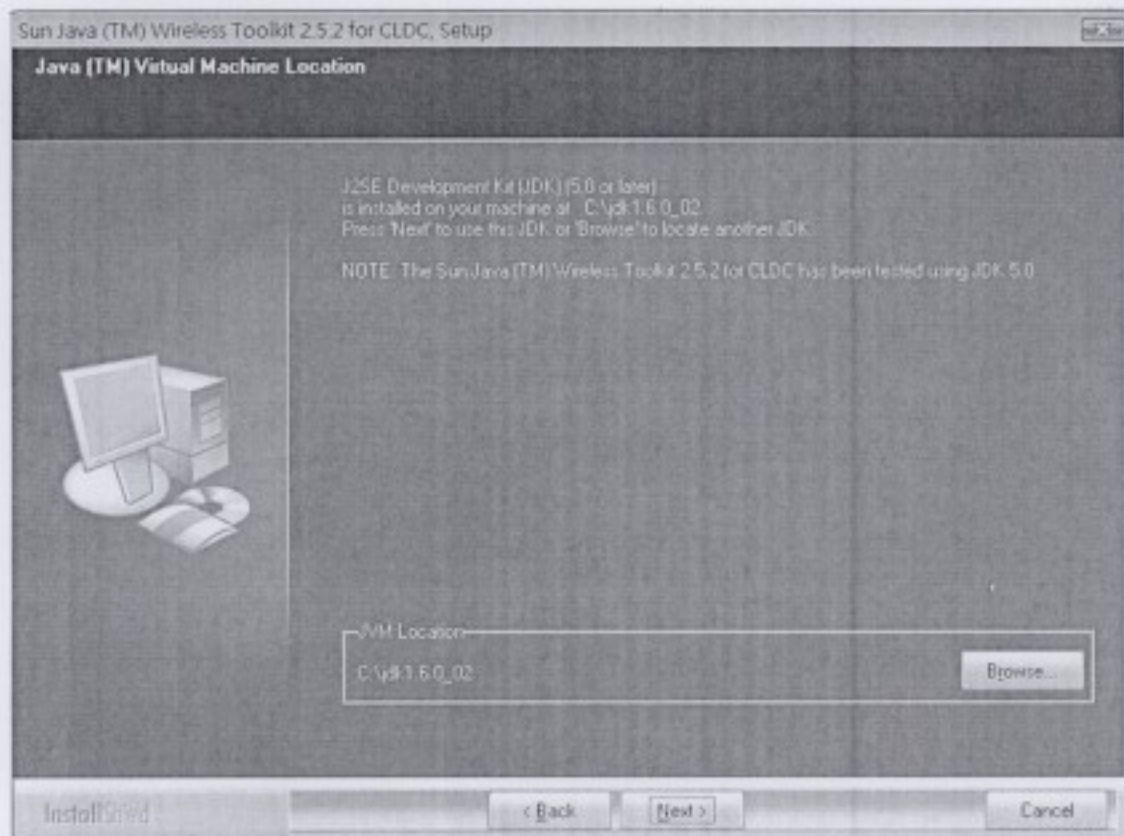
<http://java.sun.com/products/sjwtoolkit/download.html>

http://java.sun.com/products/sjwtoolkit/zh_tw_download-2_5_2.html

2. Sun Java Wireless Toolkit (先前稱為 J2ME Wireless Toolkit) 是一組建立 Java 應用程式的工具，這些應用程式可在與 Java Technology for the Wireless Industry (JTWI, JSR 185) 規格以及 Mobile Service Architecture (MSA, JSR 248) 規格相容的裝置上執行。這組工具包含生成工具、公用程式和裝置模擬器。

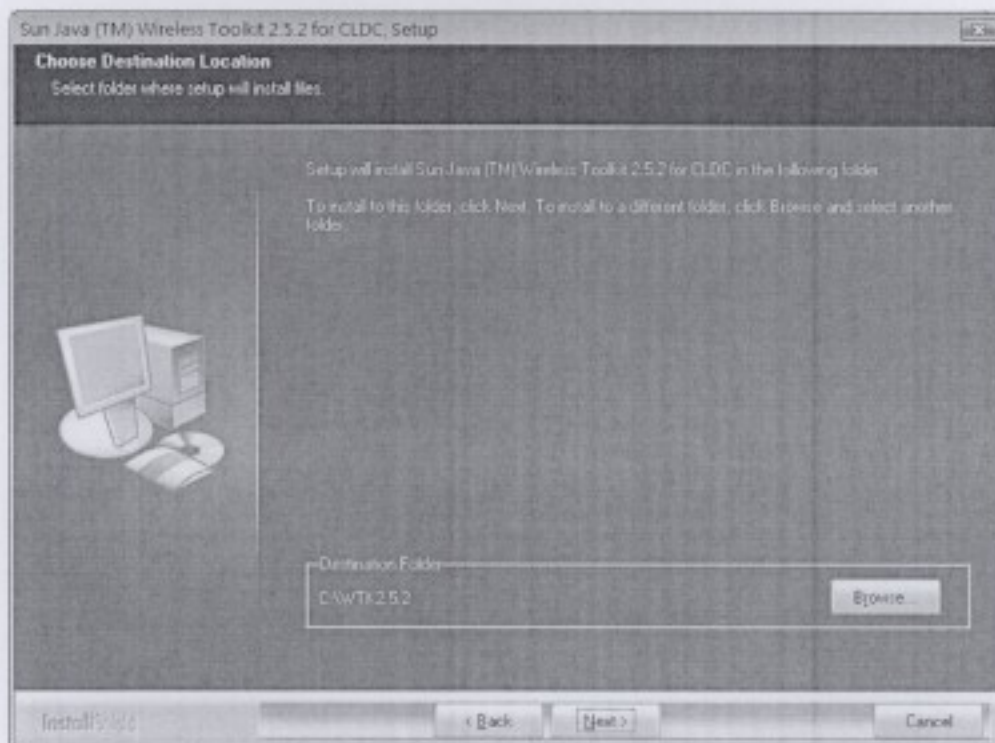
■ 下載 `sun_java_wireless_toolkit-2_5_2-windows.exe`，及執行安裝。

■ 指定 Java 之所在 JDK 目錄，例如: `c:\jdk1.6.0_02`

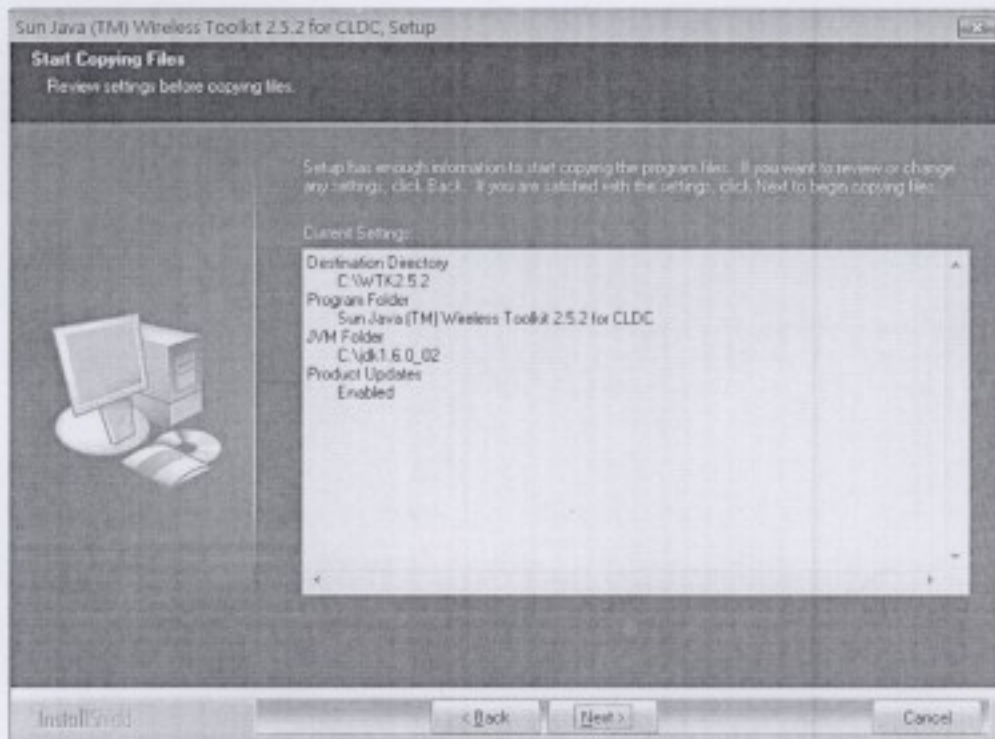


5-3. 設定 Sun Java Wireless Toolkit for CLDC 之安裝目錄

例如: c:\WTK2.5.2



■ 安裝設定確認



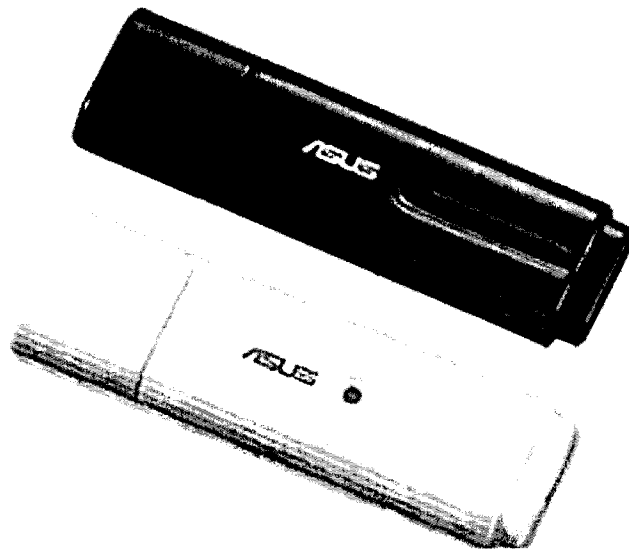
■ Apple QuickTime 播放器

下載網址:

<http://www.apple.com/quicktime/download/>

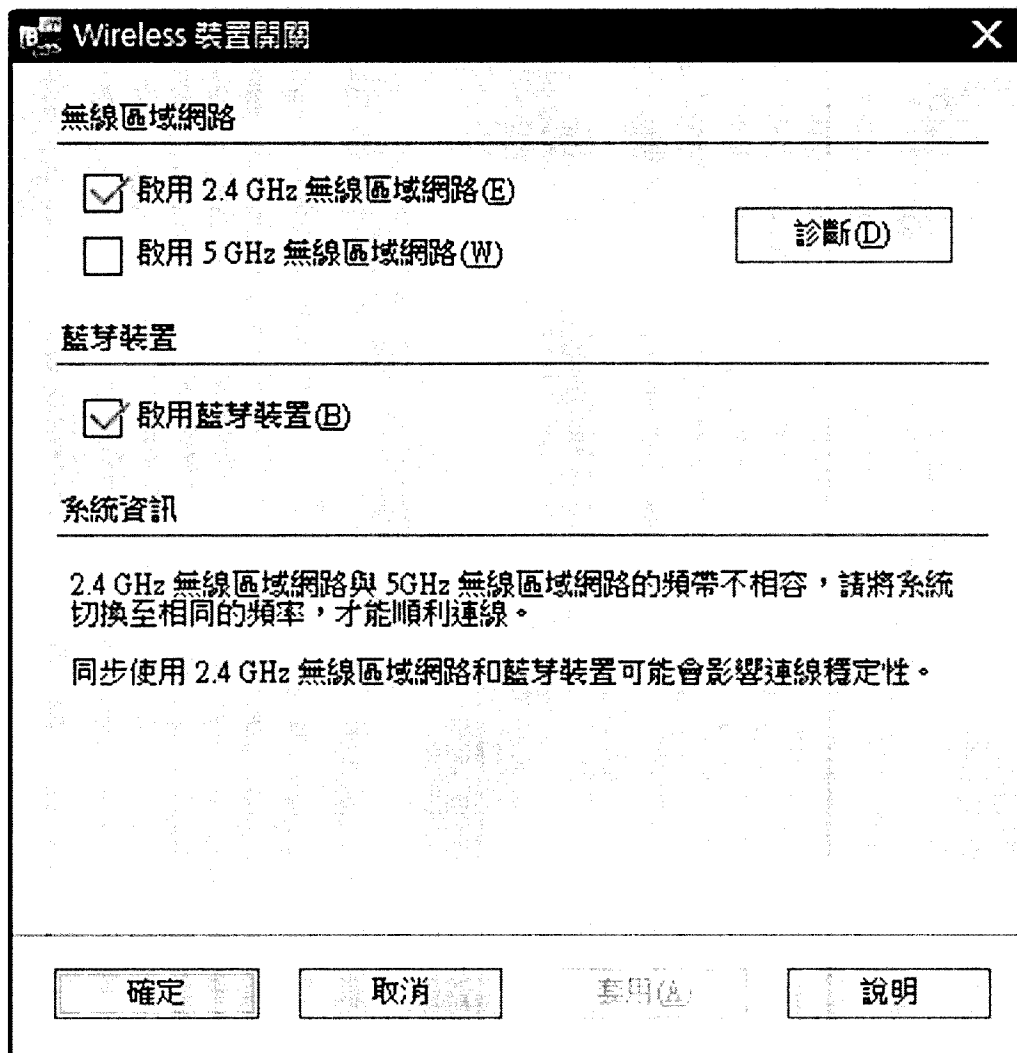
在 Windows 上播放 AMR 媒體時需要, 請下載 QuickTimeInstaller.exe, 並執行安裝。

■ 藍芽接收器



若 Notebook 原本已有內建, 可以直接利用, 若沒有則請接著安裝藍芽接收器之驅動程式。

安裝完成後, 接著將藍芽通訊打開。



5-4. Mobile Processing IDE

下載網址:

<http://mobile.processing.org/download/mobile-0006-expert.zip>

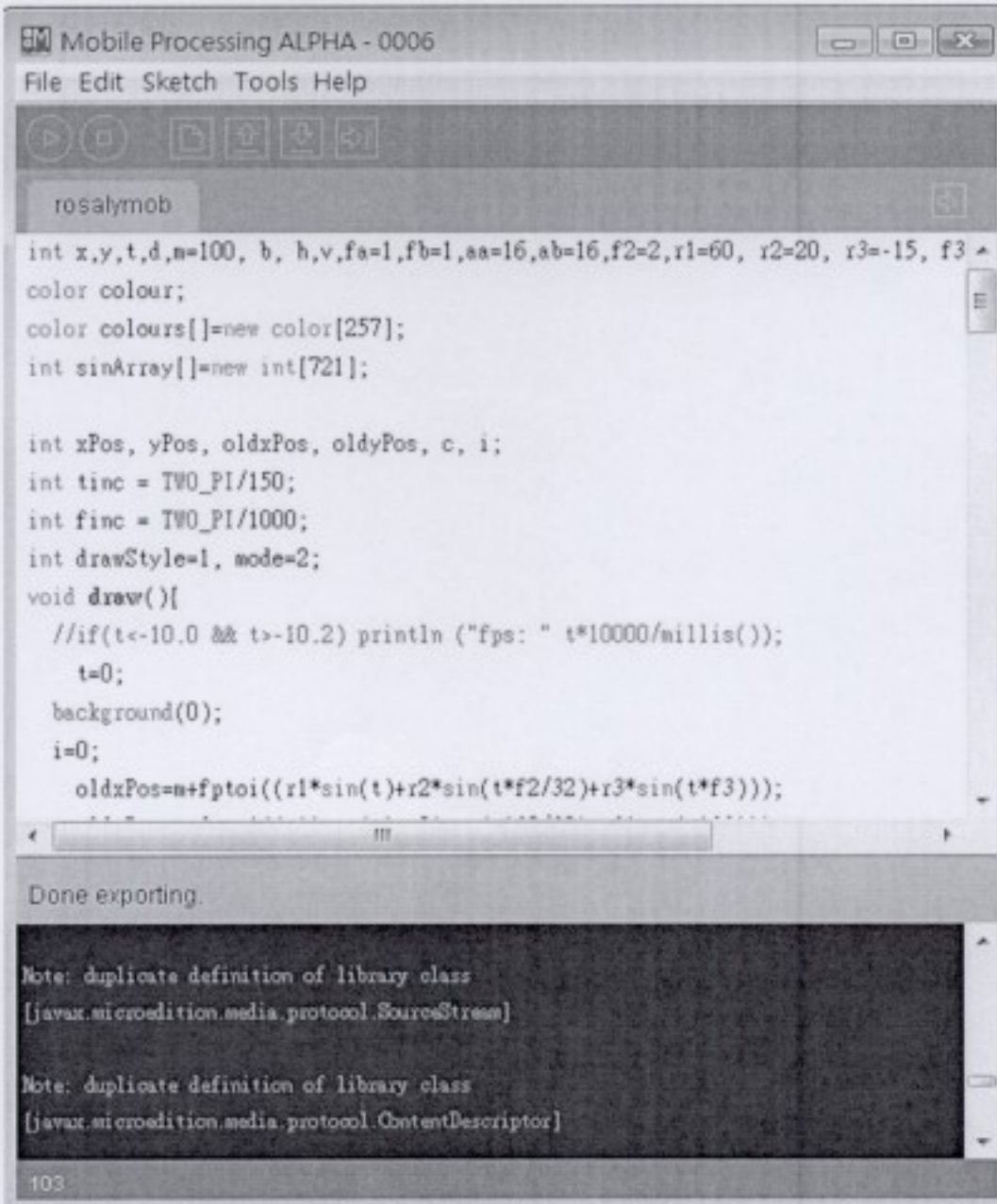
Run Mobile Processing. Choose Preferences from the main drop-down menu. In the Preferences dialog box, go to the Mobile tab, and enter the location of the WTK.

From the Help menu, choose Check for updates... to check for and automatically download the latest versions of the core API and libraries over the Internet.

Documentation

<http://mobile.processing.org/download/docs.zip>

1. 請將 mobile-0006-expert.zip 解壓縮至 c:\mobile-0006-expert 目錄下，執行 mobile.exe 開啓編輯環境。



```
Mobile Processing ALPHA - 0006
File Edit Sketch Tools Help

rosalymob

int x,y,t,d,n=100, b, h,v,fa=1,fb=1,aa=16,ab=16,f2=2,r1=60, r2=20, r3=-15, f3
color colour;
color colours[]=new color[257];
int sinArray[]=new int[721];

int xPos, yPos, oldxPos, oldyPos, c, i;
int tinc = TWO_PI/150;
int finc = TWO_PI/1000;
int drawStyle=1, mode=2;
void draw(){
  //if(t<-10.0 && t>-10.2) println ("fps: " t*10000/millis());
  t=0;
  background(0);
  i=0;
  oldxPos=n+fptoi((r1*sin(t)+r2*sin(t*f2/32)+r3*sin(t*f3)));
}

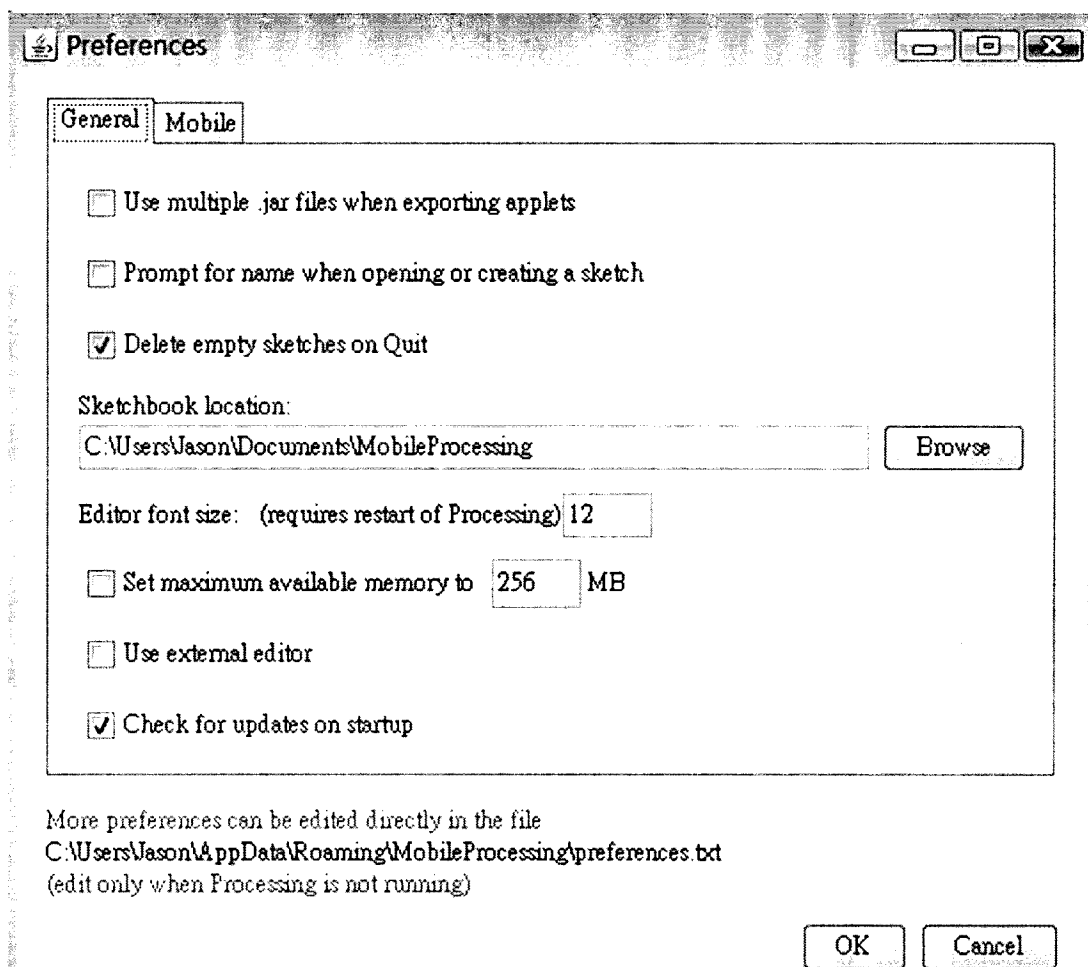
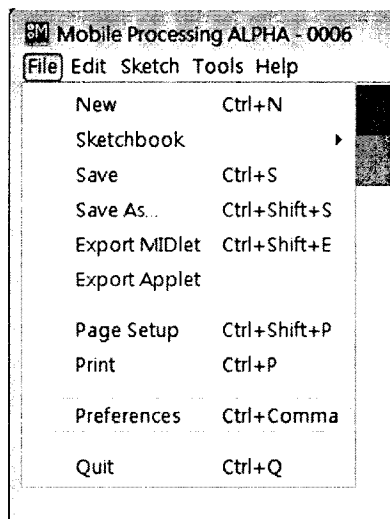
Done exporting.

Note: duplicate definition of library class
[javax.microedition.media.protocol.SourceStream]

Note: duplicate definition of library class
[javax.microedition.media.protocol.ContentDescriptor]

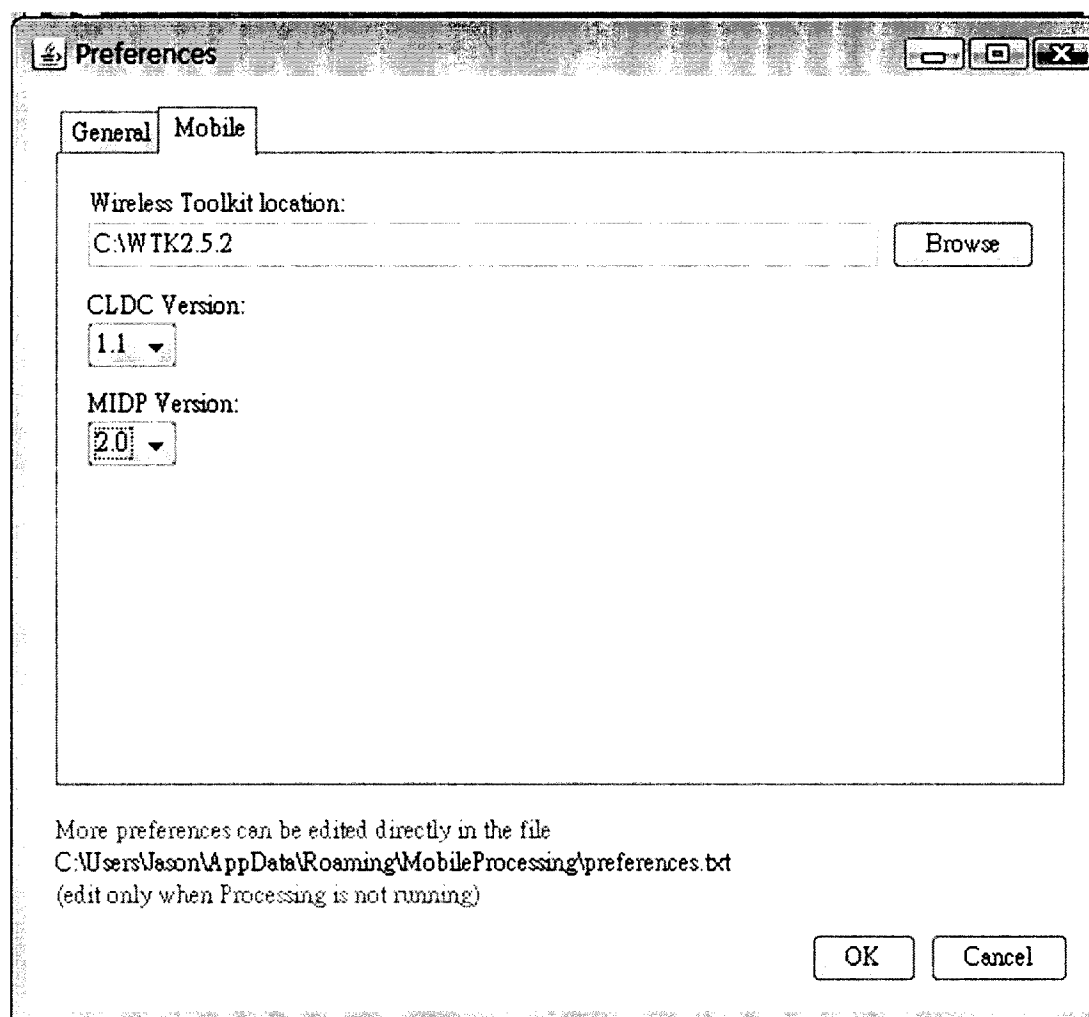
103
```

2. 設定工作目錄 (Sketchbook location), 點選 File-> Preferences 選項, 在 Preferences 對話框中, 在 General Tab 中設定 Sketchbook location, 例如: C:\Users\Jason\Documents\MobileProcessing,



More preferences can be edited directly in the file
C:\Users\Jason\AppData\Roaming\MobileProcessing\preferences.txt
(edit only when Processing is not running)

3. 設定 Wireless Toolkit location, 在 Preferences 對話框中, 在 Mobile Tab 中設定 Wireless Toolkit location, 例如: C:\WTK2.5.2, 以及設定 CLDC Version 爲 1.1, MIDP Version 爲 2.0。



5-5. 手機 Java ME JSR 程式庫支援程度檢查

<http://mobile.processing.org/phones/index.php>

<http://mobile.processing.org/phones/list.php>

1. 點選其中一個手機型號
以 Nokia N73 為例



Nokia N73

Configuration	CLDC-1.1
Profiles	MIDP-2.0
Screen (w x h)	240 x 234
Fullscreen (w x h)	240 x 320
Colors	16777216
Alpha (levels)	256
Library compatibility	Bluetooth yes
	Image2 yes
	Messaging yes
	Phone yes
	Sound yes
	Video (playback) yes
	Video (snapshot) yes
	XML yes

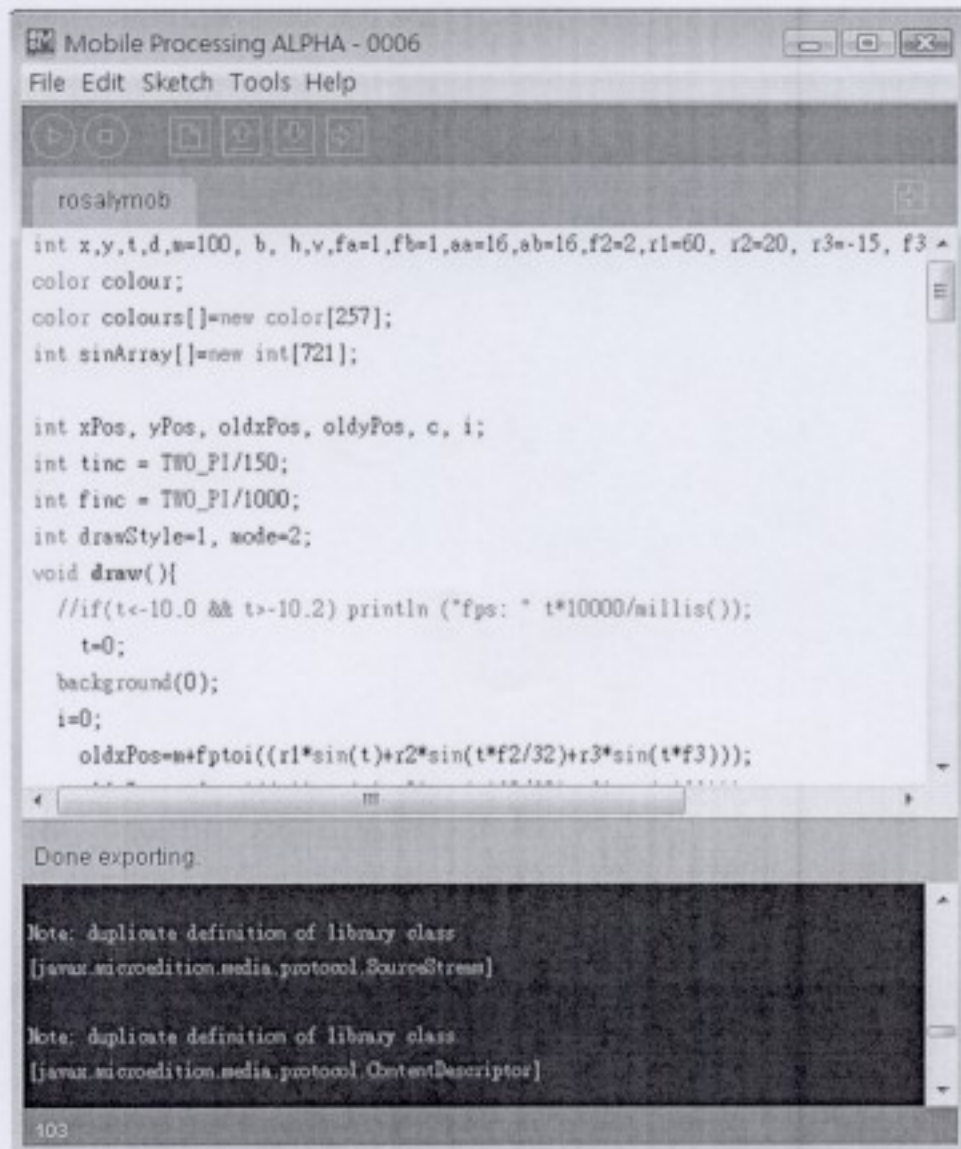
Multimedia	MMAPI Version	1.1
	Supports Mixing	yes
	Audio Capture	yes
	Video Capture	yes
	Recording	yes
	Video Encodings	video/3gpp video/3gpp2 video/mp4
	Audio Encodings	audio/wav audio/x-wav audio/basic audio/x-au audio/au audio/x-basic audio/amr audio/mp4 pcm ulaw gsm
	Snapshot Encodings	png jpeg image/png image/jpeg image/jpg image/gif image/bmp
	Streaming	None

第六章、Mobile Processing 開發程序

【Mobile 與多媒體創作】工作坊

Mobile Processing

1. 開啓 Mobile Processing 編輯環境



2. 程式碼編寫

3. 利用手機模擬器，執行及測試程式。







4 程式編譯及匯出成 MIDlet。

4.1 以下是相關程式說明，以 `rosalymob` 為例
`rosalymob.pde` → Mobile Processing 原始碼
`rosalymob.java` → Java 原始碼

rosalymob.jad → 手機應用程式描述檔 MIDP Java Application Descriptor

rosalymob.jar → 手機應用程式執行檔

其中 rosalymob.jad, rosalymob.jar 是要下載至手機中執行。

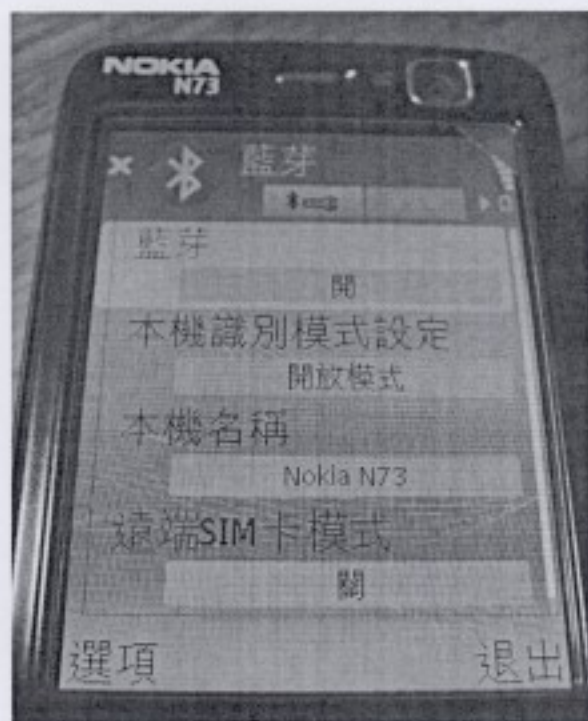
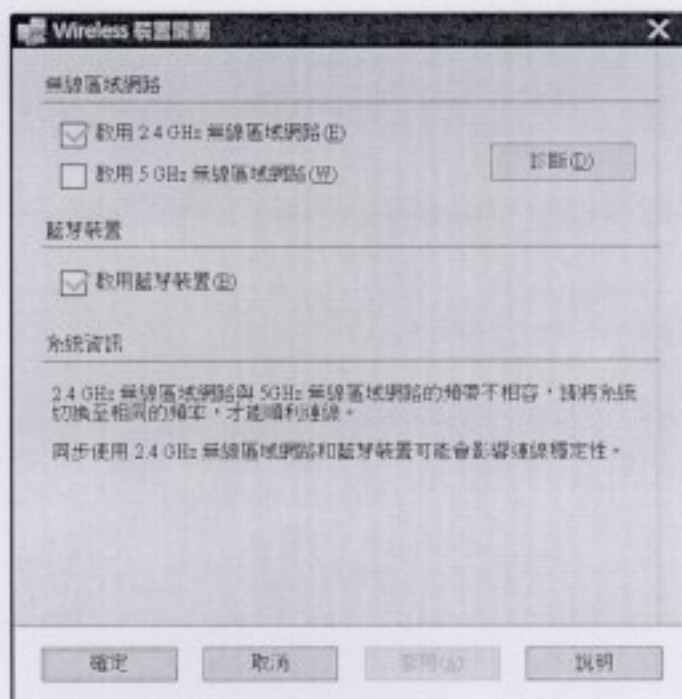
名稱	修改日期	類型	大小
 rosalymob.jad	2007/10/23 下午 10...	MIDP Java Application Descriptor	1 KB
 rosalymob.jar	2007/10/23 下午 10...	Executable Jar File	44 KB
 rosalymob.java	2007/10/23 下午 10...	JAVA 檔案	3 KB
 rosalymob.pde	2007/10/23 下午 10...	PDE 檔案	3 KB

4.2 以下是 rosalymob.jad 的內容

```
MIDlet-1: rosalymob, /icon.png, rosalymob
MIDlet-Icon: /icon.png
MIDlet-Name: rosalymob
MIDlet-Version: 1.0.0
MIDlet-Vendor: Mobile Processing ALPHA - 0006
MicroEdition-Profile: MIDP-1.0
MicroEdition-Configuration: CLDC-1.0
MIDlet-Jar-URL: rosalymob.jar
MIDlet-Jar-Size: 44635
```

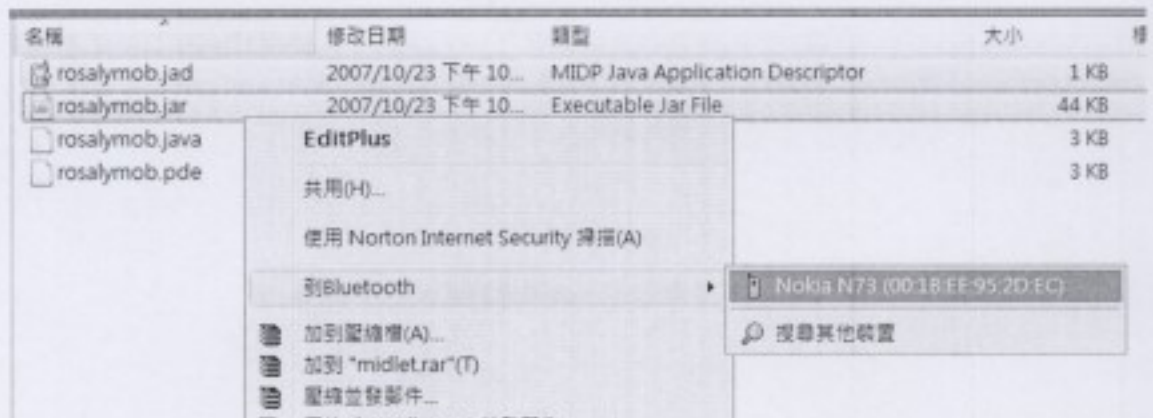
5. 傳輸應用程式至手機

5.1 將電腦及手機的藍芽通訊打開，並在電腦端執行藍芽裝置搜尋及連線配對。在進行連線配對時，在電腦及手機的畫面上會出現要求輸入密碼。





5.2 利用藍芽將 `rosalymob.jad`, `rosalymob.jar` 傳輸至手機，傳輸完成後會儲存在什麼位置或目錄，會因不同的手機而有不同位置，以 **Nokia N73** 為例，會先儲存在“訊息的收件匣”中。

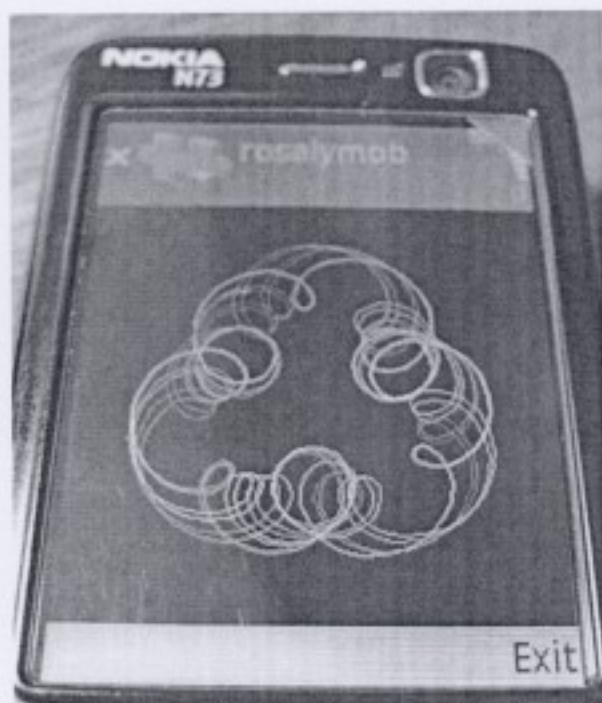




5.3 安裝程式，點選收件匣中的 rosalymob.jad 就會啟動安裝程式程序。



5.4 執行程式，安裝後，程式會放置在“應用程式”中，會看到 rosalymob 的圖示，點選後即可執行程式。(不同的手機會放置在不同位置)



5.5 解除程式，將選取位置移至 rosaly mob 的圖示上，按下"選項"，點選"移出"即可。

5.6 刪除程式，回到訊息的收件匣，選取 rosaly mob.jad，按下"選項"，點選"刪除"即可。