

Computer Organization and Assembly Languages Final Project

Kill Bricks

B94902007 張詩平

B94902081 陳秋語

遊戲簡介

這款「Kill Bricks」，其實就是有二十多年歷史的「打磚塊遊戲」。從傳統的遊戲機、電視遊樂器到今日的網路遊戲，打磚塊一直是歷久不衰的小遊戲，以其簡單易上手的特質與豐富的變化性，長久以來受到眾人的喜愛。他是一款動作類型的遊戲，玩家必須操作一根短短的橫棒，撞擊落下的小圓球來打碎上方的磚牆，並避免球掉到橫棒下方，將所有磚牆消去即可獲得勝利。

打磚塊遊戲起源於美國 Atari 公司於 1976 年推出的街機遊戲「Breakout」，是由該公司 1972 年發行的「PONG」改良而來。「PONG」是世界上第一款電子遊戲，類似桌球的玩法，雙方各持有一光棒來回打擊光點，將光點打至對方背後的球門，就得到此局的勝利。相較於「PONG」，「Breakout」多了可以一人遊戲以及變化豐富的優勢，馬上造成玩家們的大轟動，流傳到世界各地。

至今科技的發達與網路的普及，我們可以輕易的見到各種類型的打磚塊遊戲，從基本規則到簡單的變化（速度、橫棒長度、橫棒可以發射子彈等功能），種類之多不勝枚舉，更有不少因他而衍生的小遊戲誕生。由於他的簡單易玩，現今很多行動電話也可以看到他的身影，普遍性之高想見一般。

這次我們所設計的「Kill Bricks」，真實的呈現打磚塊遊戲的原始精神，配合幾款精心設計的小關卡，在簡單中見到變化，變化又不失簡單，讓玩家藉由這款輕鬆的小遊戲，回味過去打磚塊的趣味，體會打磚塊遊戲迷人之處與精髓之所在。

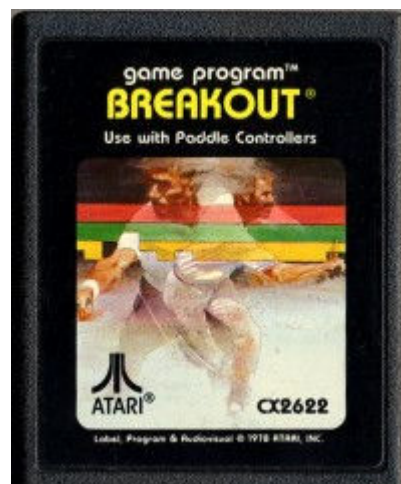


Figure1. Atari's Breakout



Figure2. Breakout's interface

設計概述

遊戲設計仿傳統打磚塊遊戲的慣例，有三顆球的生命值，每擊中一個磚塊即得到十分，破完一關可獲得五百分的 bonus，在有限的球內把磚塊擊完則可進入下個關卡。本遊戲設計兩個關卡，分別是基本款（矩形的磚塊組合）與進階版（紫色磚塊必須擊中兩次方可消去），在生命值尚有殘餘的情況下破完兩關，將獲得最後的勝利。



Figure3. welcome page



Figure4. level 1



Figure5. level 2

在遊戲過程中，玩家隨時可按空白鍵暫停遊戲，或是按 ESC 鍵結束遊戲，當玩家不幸用完三顆球的生命值，可以選擇再度挑戰或是離開遊戲，全部都破關時也可以決定是否再玩一次，玩家如果選擇結束遊戲，則會進入結尾的小動畫。



Figure6. end page

為了遊戲設計者執行上的順利以及替遊戲增添神秘的樂趣，我們設計了一點小密技幫助破關，在遊戲進行中，按下 ESC 鍵會出現 ESC (Y/N) 字幕，此時按下 G 鍵會自動進入下一關，並將破完前一關應得的分數加入 score 中，讓玩家可以繼續挑戰下一關而不損失應得的分數。

遊戲中搭配了豐富的顏色，讓玩家在休閒的同時也多了視覺的享受，另外，在擊中磚塊的同時也搭配了音效，加強打中磚塊的真實感。遊戲裡不管是開頭、結尾、晉級或是全部破關時，都配上不同的字幕，給玩家不一樣的驚奇，鼓勵玩家勇敢面對「Kill Bricks」強大的挑戰，也讓遊戲者在破關的時刻，重溫記憶中打磚塊遊戲的舊夢。



Figure7. Good Job!

在此提供按鍵對照表，供諸位玩家參考：

按鍵	功能
←	將橫棒向左移
→	將橫棒向右移
PAUSE	暫停遊戲 (任意鍵回到遊戲)
ESC	進入"是否離開遊戲"畫面
按下 ESC 後,進入 ESC(Y/N)頁面	
Y	結束遊戲
N	回到遊戲
G	進入下一關 (最末關進入結束畫面)
在 Play Again(Y/N)頁面時	
Y	重新開始遊戲
N	結束遊戲

實作內容

「Kill Bricks」的架構不會過於繁雜，因此我們選擇全部由組語來完成。概括來說，整個程式是由一個中心的主體架構，加上許許多多不同功能的函式結合而成。有負責印螢幕的 SetScreen，將起始狀態和每一個關卡印出；有負責判斷輸入的 CheckInput，根據輸入值回傳對應的動作；movBar 與 movBall，顧名思義，用他們來做出橫棒和球的移動；Rebound 用來判斷當球遇到障礙的時候應該如何反彈，其中包括了遇到牆面的反彈以及遇到磚塊時必須消掉磚塊再反彈，在這裡我們使用判斷顏色的方法，根據碰撞到的顏色來決定應該做的反應，紅磚或紫磚有不同的碰撞結果；reboundBricks 特別對碰到磚塊的反彈作更多的動作；reboundDirect 更仔細的考慮小球反彈橫棒後的方向，讓球的路徑更多樣化；WriteString 可以寫入字串；Draw 用以完成畫圖的部分；而 DrawBricks 則可將磚塊一個個畫好，做出每一個關卡。

此外還有其餘大小函式相互輔助，讓程式撰寫變得清楚簡潔，我們需要用到各種功能時，只需要呼叫每一個函式，不必一一寫出，不但解省時間，又增加程式的可讀性，也應用了物件導向的概念。

組合語言的特色便是可使程式加速，當我們使用高階語言，在上述一連串呼叫函式的過程中，可能會多了不必要的步驟而多耗費時間，使用組合語言，以撰寫時比較困難的低階語言來換取程式的快速，在部分程式的加速時是十分常見的。

縱使在資訊發展快速的現代，組語依舊有他的重要之處，這也是為何有了其他高階語言，我們依舊要鑽研組合語言的學問。當然，這也激勵了更多的程式設計師，往研發更便利且快速的語言這方向來努力。



Figure8. 電腦或手機上的打磚塊遊戲

參考資料

Textbook -- Assembly Language for Intel-Based Computers

Wikipedia-- Breakout

<http://en.wikipedia.org/wiki/Breakout>

CNET-- <http://gamecenter.taiwan.cnet.com/News/20001229/2000122904.html>