|  |
| --- |
| 東吳大學 資訊管理學系 |
| XNA 2D遊戲程式設計Throw Fish In |
| 指導教授 曾修宜 副教授 |
|  |
| 96156133陳映潔 |
| 99156235吳致芳 |
| **2013/12/25** |

目錄

[**壹、** **摘要** 2](#_Toc376430273)

[**貳、** **系統規格概述** 3](#_Toc376430274)

[ 遊戲名稱 3](#_Toc376430275)

[ 遊戲設計目的 3](#_Toc376430276)

[ 使用環境 3](#_Toc376430277)

[ 程式語言 3](#_Toc376430278)

[ 繪製圖形 3](#_Toc376430279)

[ XNA開發環境需求 3](#_Toc376430280)

[**參、** **原理與方法** 3](#_Toc376430281)

[ 遊戲設計概述 4](#_Toc376430282)

[ 製作步驟 4](#_Toc376430283)

[ 架構圖 6](#_Toc376430284)

[**肆、** **遊戲設計** 7](#_Toc376430285)

[ 遊戲運作流程 7](#_Toc376430286)

[ 執行畫面 8](#_Toc376430287)

[遊戲主選單 8](#_Toc376430288)

[遊戲進行畫面 8](#_Toc376430289)

[About畫面 10](#_Toc376430290)

[ 開發過程遭遇的困難 10](#_Toc376430291)

[**伍、** **參考資料** 11](#_Toc376430292)

[**陸、** **附錄** 12](#_Toc376430293)

[XNA之介紹 12](#_Toc376430294)

# **摘要**

遊戲產業是目前國內數位內容發展當中的重點產業，尚未有關拖拉物件後沿著拋物線拋出的遊戲，Throw Fish In操作簡單卻能讓玩家手忙腳亂，挑戰運動神經與肢體，希望能使玩家藉由操作此遊戲訓練手眼協調與腦並用以及達到娛樂效果。

# **系統規格概述**

## 遊戲名稱

　　Throw Fish In

## 遊戲設計目的

希望透過滑鼠或是觸控螢幕在大賣場裡供全家一起玩樂，不僅能帶來歡樂且訓練手眼協調與腦並用的操作能力。

## 使用環境

　　Microsoft Visual Studio 2010、XNA

## 程式語言

　　C#

## 繪製圖形

　　Photoshop CS4

## XNA開發環境需求

　　作業系統：

　　　　在Windows XP，或是之後的版本

　　硬體配備：

　　　　具有支援Shader版本1.1版或以上的3D加速顯示繪圖晶片

　　　　至少1Ghz以上的CPU

　　　　至少192MB的記憶體

　　軟體元件：

　　　　.Net Framework 2.0或更新的版本

　　　　DirectX 9.0c 或更新的版本

　　　　XNA Framework

# **原理與方法**

## 遊戲設計概述

　　本專題所設計的遊戲類型以休閒遊戲（Casual game）為主軸。是一種易於上手，不涉及短時間大量或重度腦力活動，規則相對簡單的電腦遊戲，部分考驗玩家的操作技巧和精神集中力。遊戲畫面中有一個魚缸，不停會有魚從畫面兩側出現並游動，為了達到過關標準，必須要將魚放入或拋進魚缸，來使得遊戲順利進行，若不幸在時間截止前未達到過關分數，那麼遊戲就結束了。若是放入特殊魚種，則可得到獎勵分數，可藉此來獲得高分；不幸抓到特定魚種則會倒扣分數。遊戲的樂趣，就是在時間結束之前，盡力去提高分數。

## 製作步驟

1. 本專題遊戲由「Microsoft Visual Studio 2010」軟體製作，語言為中文，此軟體試用版可在Microsoft官網合法下載。另外從 MSDN® (網址為：msdn2.microsoft.com/xna) 下載安裝 XNA Game Studio Express。
2. 先設計魚游動的分解動作。(如圖一)

|  |
| --- |
|  |
| 圖一　魚游動分解圖 |

1. 設計扣分魚的游動分解動作。(如圖二)

|  |
| --- |
|  |
| 圖二　扣分魚游動分解圖 |

1. 載入要呈現遊戲的圖檔，將圖檔匯入專案，撰寫程式繪製出背景、物件（魚缸及魚）。 (如圖三)

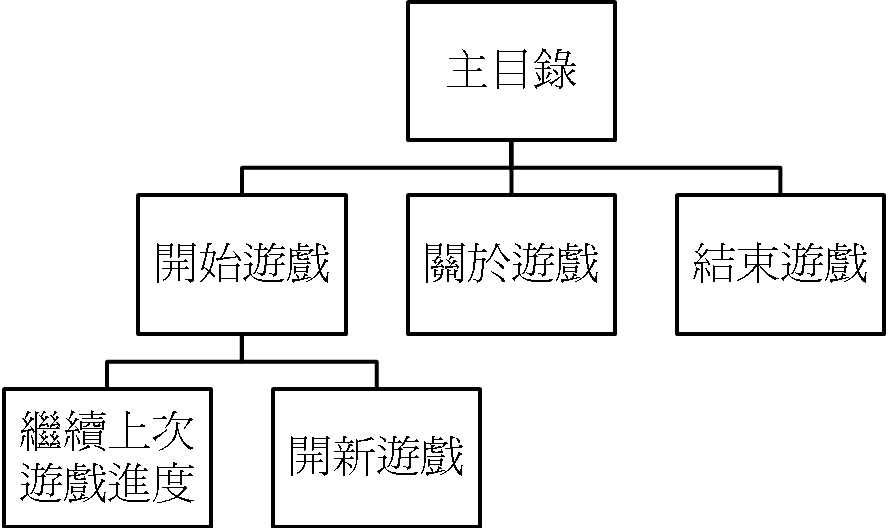
|  |
| --- |
|  |
| 圖三　繪製出背景、物件 |

1. 撰寫程式將拖拉物件功能加入，接著加上釋放物件後拋物線移動的效果。
2. 增加碰撞魚缸判斷，如果魚碰撞到魚缸上緣要讓牠落下後最後停在魚缸裡，否則落在魚缸外。
3. 處理遊戲中所有元件角色的邏輯條件，如魚丟入魚缸內，分數加100或是扣100。
4. 加入倒數計時功能，時間歸零遊戲結束。達到目標分數贏得遊戲，否則失敗。
5. 增加音效，配合遊戲的進行音效的增加，使遊戲的環境更有變化性。
6. 增加關卡數。
7. 繪製遊戲介面及遊戲選單，撰寫遊戲選單頁面程式並合併遊戲主程式。執行遊戲時，會進入遊戲選單(如圖四)，用滑鼠或觸控螢幕觸控選擇項目可進入遊戲畫面(如圖五)。

|  |
| --- |
|  |
| 圖四 遊戲選單 |
| Start：遊戲開始 |
| About ：遊戲操作簡介及製作團隊 |
| Exit：離開遊戲（關閉視窗） |
|  |
| 圖五 進入畫面 |

1. 測試遊戲的進行，並修改遊戲進行時所發生的錯誤，將錯誤修改至最小。

## 架構圖



# **遊戲設計**

## 遊戲運作流程

|  |
| --- |
|  |
| 圖六　遊戲運作流程 |

1. Game1( ) 一般初始化 → Initialize( )遊戲初始化
2. LoadContent( )
3. Update(GameTime gameTime)
4. Draw(GameTime gameTime)
5. UnloadContent( )

|  |
| --- |
|  |

# 執行畫面

## 遊戲主選單

|  |
| --- |
|  |
| 圖七　遊戲主選單 |

## 遊戲進行畫面

|  |
| --- |
|  |
| 圖八　關卡一 |
|  |
| 圖九　關卡二 |
|  |
| 圖十　關卡三  圖十一 遊戲成功  圖十二 遊戲失敗 |

## About畫面

|  |
| --- |
|  |
| 圖十三　About選單 |

# 開發過程遭遇的困難

原本使用的開發軟體為Adobe Flash Professional CS6，但是後來因為開發環境的限制（在處理多點觸控時，無法判別單一物件的id並且指定其移動，導致所有物件一起動作），再加上Ｆlash（.swf）會因機器、設備的不同而發生讓程式無法動作的情況，所以將開發環境和程式語言更改為Microsoft Visual Studio 2010 C#、XNA。

程式方面，拋物線要計算每隻魚被拖曳的距離、時間、位置，並且要用多工的方式，避免第一隻魚還沒完成拋物線動作，程式還未執行完成，就開始進行第二隻的拋物線動作。為了讓每隻魚栩栩如生，所以利用RANDOM讓每隻魚上下左右移動的偏移量不同。至於遊戲架構，換頁是採用參數的方式，呼叫出不同的畫面，為了讓使用者能回到主選單畫面或回到上一頁，所以在呼叫下一個頁面時，必須先Remove( )當前畫面，再Change( )新的畫面，避免參數傳遞錯誤。

# **參考資料**

鄞永傳（2011）。《XNA 4遊戲程式設計：適用Windows PC、Phone、Xbox 360》。台北：松崗。

XNA 簡介 - Microsoft Download Center。2013．12．23取自<https://www.google.com.tw/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDgQFjAB&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F0%2Fc%2Fc%2F0cc71b79-0447-4310-863e-0435515484d3%2FCG004400_ch1.pdf&ei=80C4UovwDYzJkwWL9YHYBA&usg=AFQjCNFmX-G4YWfy2BHJiTwwaHaaZQZ0YQ&sig2=6hsHx-YWmCA0Mgie8vuuAA>。

Follow Fang!: XNA 4.0 簡介。2013．12．23取自<http://writecodepeople.blogspot.tw/2013/09/xna.html>

# **附錄**

## XNA之介紹

　　XNA Game Studio Express 採用 Visual C#® 2005 Express Edition 為基礎，建置在 XNA Framework 之上，這是一個針對遊戲開發而設計的 Managed 程式碼類別庫。此架構簡化了為 Xbox 360 與 Windows XP SP2 這兩個差異極大的平台撰寫程式碼的程序。實際上大部分所撰寫的程式碼都能在這兩個系統上執行，而且完全不需更動。另外也能在Windows Phone平台執行。

　　XBOX 360以及Windows和Windows Phone三種產物的硬體規格皆不相同，XNA為了解決該問題，XNA提供兩種格式，分別是Reach和HiDef，以下表格將詳細說明Reach和HiDef的差異：

|  |  |  |
| --- | --- | --- |
|  | **Reach** | **HiDef** |
| *Supported platforms* | Windows Phone 7 Series, Xbox 360, and any Windows PC with a DirectX 9 GPU that supports at least shader model 2.0 | Xbox 360, and any Windows PC with a DirectX 10 (or equivalent: see below) GPU |
| *Shader model* | 2.0  (but Windows Phone does not support custom shaders) | 3.0+  (Xbox 360 supports custom shader extensions such as vfetch, which are not available on Windows) |
| *Max texture size* | 2048 | 4096 |
| *Max cubemap size* | 512 | 4096 |
| *Max volume texture size* | Volume textures are not supported | 256 |
| *Non power of two textures* | Conditional: cannot use wrap addressing mode, mipmaps, or DXT compression when the size is not a power of two | Yes |
| *Non power of two cubemaps* | No | Yes |
| *Non power of two volume textures* | Volume textures are not supported | Yes |
| *Max primitives per draw call* | 65535 | 1048575 |
| *Index buffer formats* | 16 bit | 16 and 32 bit |
| *Vertex element formats* | Color, Byte4, Single, Vector2, Vector3, Vector4, Short2, Short4, NormalizedShort2, NormalizedShort4 | All of the Reach formats, plus HalfVector2, HalfVector4 |
| *Texture formats* | Color, Bgr565, Bgra5551, Bgra4444, NormalizedByte2, NormalizedByte4, Dxt1, Dxt3, Dxt5 | All of the Reach formats, plus Alpha8, Rg32, Rgba64, Rgba1010102, Single, Vector2, Vector4, HalfSingle, HalfVector2, HalfVector4. Floating point texture formats do not support filtering. |
| *Vertex texture formats* | Vertex texturing is not supported | Single, Vector2, Vector4, HalfSingle, HalfVector2, HalfVector4 |
| *Render target formats* | Variable (see below) | Variable (see below) |
| *Multiple render targets* | No | Up to 4. Must all have the same bit depth. Supports alpha blending and independent write masks per rendertarget. |
| *Occlusion queries* | No | Yes |
| *Separate alpha blend* | No | Yes |
| *Blend.SourceAlphaSaturation* | Only for SourceBlend, not DestinationBlend | Yes |
| *Max vertex streams* | 16 | 16 |
| *Max stream stride* | 255 | 255 |

　　XNA Game Studio Express是建置在 Visual C# 2005 Express Edition 之上的遊戲開發平台，使用 C# 與 Visual C# 2005 Express Edition 整合式開發環境，同時在 Windows 和 Xbox 360 上建造自己的遊戲。包含 XNA Framework，XNA Framework 是 Windows 與 Xbox 360 都適用的跨平台遊戲開發 API，它不僅能大幅簡化圖形、音訊、輸入和儲存，還能改進所有遊戲最根本的計時和繪圖迴圈。