|  |
| --- |
|  |
| 物件導向程式設計 |
| 指導老師：許志仲 老師 |

|  |
| --- |
| 109/06/22 |

組員：B10756008 王靜文

B10756015 王滋蘘

B10756027 李昂珊

B10756030 陳彥芝

B10756044 黃珮瑄

**目錄**

1. **研究動機 ………………………………………………………………………3**
2. **應用資料結構及原理 …………………………………………………………3**
3. **廣度優先搜尋 （BFS，Breadth-First Search） ……………………3**
4. **堆疊 (LIFO，Last In First Out) ……………………………………3**

**三、佇列 (FIFO，First In First Out)……………………………………4**

**四、雙向佇列 (Deque) ………………………………………………………4**

1. **插入排序 (Insertion Sort) …………………………………………4**
2. **A\*搜尋演算法 (A\* search algorithm) ………………………………5**

**參、程式碼關鍵部分 ………………………………………………………………5**

1. **A\*增加節點 ………………………………………………………………5**
2. **A\*佇列 ……………………………………………………………………6**
3. **A\*節點搜尋 ………………………………………………………………6**
4. **A\*雙向佇列 ………………………………………………………………7**
5. **A\*滑鼠點擊的移動位置 …………………………………………………7**
6. **A\*尋找代表的字串 ………………………………………………………8**
7. **BFS的搜尋控制 …………………………………………………………8**

**肆、報告中遇到的困難………………………………………………………………8**

**伍、成果展示…………………………………………………………………………9**

**一、程式設計介面………………………………………………………………9**

**二、執行畫面……………………………………………………………………9**

**陸、小組心得 ………………………………………………………………………10**

**一、B10756008王靜文 ………………………………………………………10**

**二、B10756015王滋蘘 ………………………………………………………10**

**三、B10756027李昂珊 ………………………………………………………10**

四**、B10756030陳彥芝 ………………………………………………………10**

五**、B10756044黃珮瑄 ………………………………………………………10**

**陸、參考資料 ………………………………………………………………………11**

**圖目錄**

**圖1.BFS 概念圖 …………………………………………………………………3**

**圖2.堆疊概念圖 …………………………………………………………………3**

**圖3.佇列概念圖 …………………………………………………………………4**

**圖4.雙向佇列概念圖 ……………………………………………………………4**

**圖5.插入排序概念圖 ……………………………………………………………4**

**圖6.A\*搜尋演算法概念圖 ………………………………………………………5**

**圖7.A\*增加節點程式碼 …………………………………………………………5**

**圖8.A\*佇列程式碼-1 ………………………………………………………………6**

**圖9.** **A\*佇列程式碼 - 2 …………………………………………………………6**

**圖10.** **A\*節點搜尋程式碼 …………………………………………………………6**

**圖11.** **A\*雙向佇列程式碼 …………………………………………………………7**

**圖12.** **A\*滑鼠點擊移動位置程式碼 ………………………………………………7**

**圖13.** **A\*尋找代表字串程式碼 ……………………………………………………8**

**圖14.** **BFS搜尋控制程式碼…………………………………………………………8**

**圖15.** **成果展示畫面-1 ……………………………………………………………9**

**圖16.** **成果展示畫面-2 ……………………………………………………………9**

1. **研究動機**

　　先前在視窗程式設計的課程中，有實作過8-Puzzle拼圖遊戲的作業，當時對於其程式碼的概念有個雛形，後來在物件導向課程中學習到BFS、佇列、排序等等演算法，才了解到原來有應用在這方面，觸使我們想更加深入，去了解它是如何運作，另外更深入分析用不同的演算法去執行，所耗的時間哪一種效率更高？所耗費的時間更少、更快速。

1. **應用資料結構及原理**
2. **廣度優先搜尋 （BFS，Breadth-First Search）**

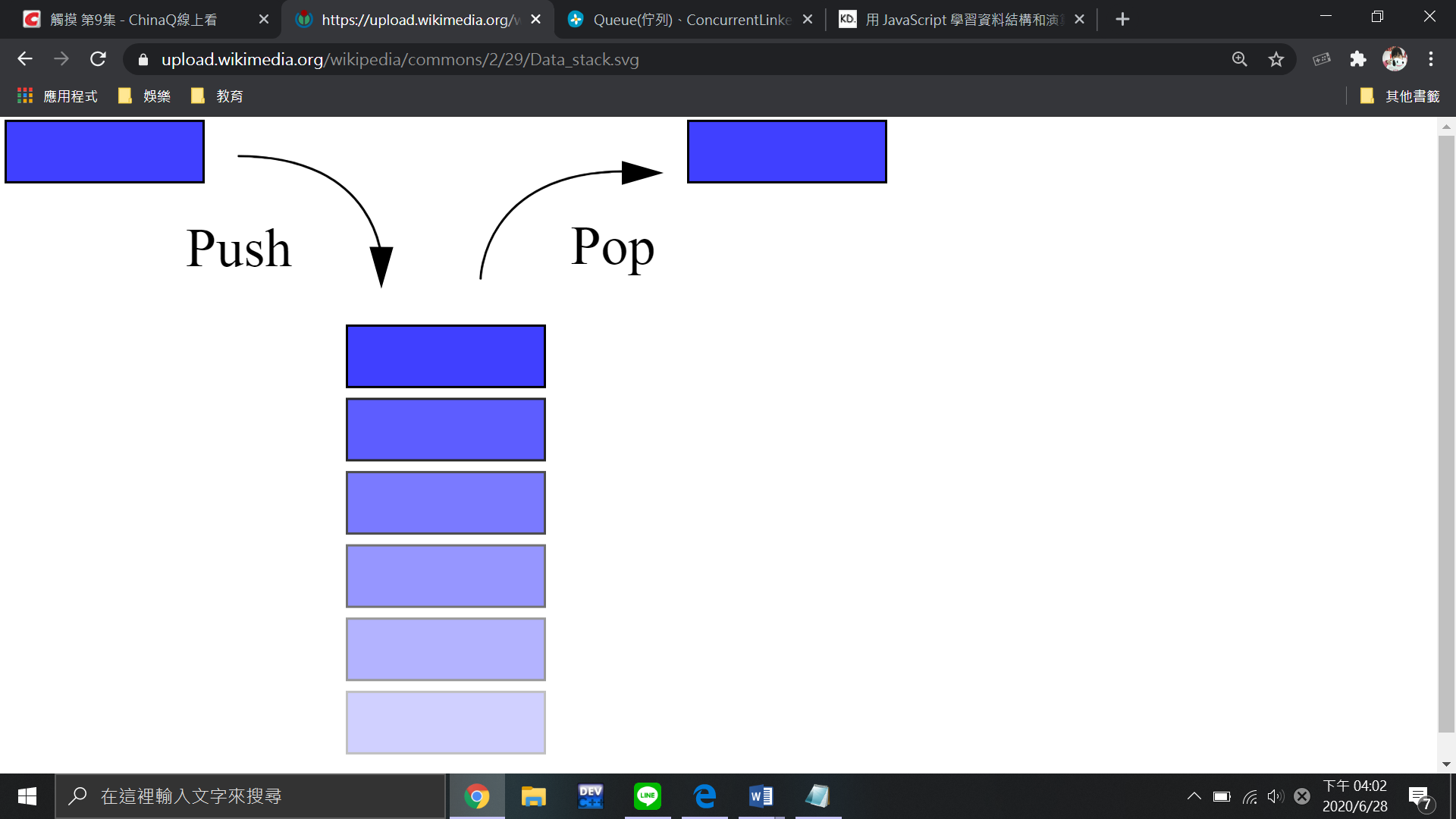
　　　一種圖形搜尋演算法，建立一個具有Vertices個數的Queue，照順序讀取資料進來，從根節點開始，不考慮結果的可能位址，系統展開並搜尋整張圖的所有節點，直到找到結果為止。如果所有節點均被存取，則演算法中止。



**圖1.BFS概念圖**

1. **堆疊 (LIFO，Last In First Out)**

　　　一種擁有特定進出規則的線性串列結構。先入後出，後入先出，除頭尾節點之外，每個元素有一個前驅，一個後繼。只允許在有序的線性資料集合的一端進行加入資料和移除資料的運算。



**圖2.堆疊概念圖**

1. **佇列 (FIFO，First In First Out)**

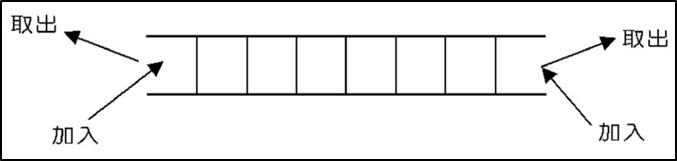
　　　佇列是一種先進先出的有序串列。在具體應用中通常用鍊表或者數組來實現。 佇列的新增和刪除元素是發生在不同端，新增元素在尾部、移除元素在頂部，最新加入的元素會從尾部排入。佇列的操作方式和堆疊類似，唯一的區別在於佇列只允許新數據在後端進行添加。

****

**圖3.佇列概念圖**

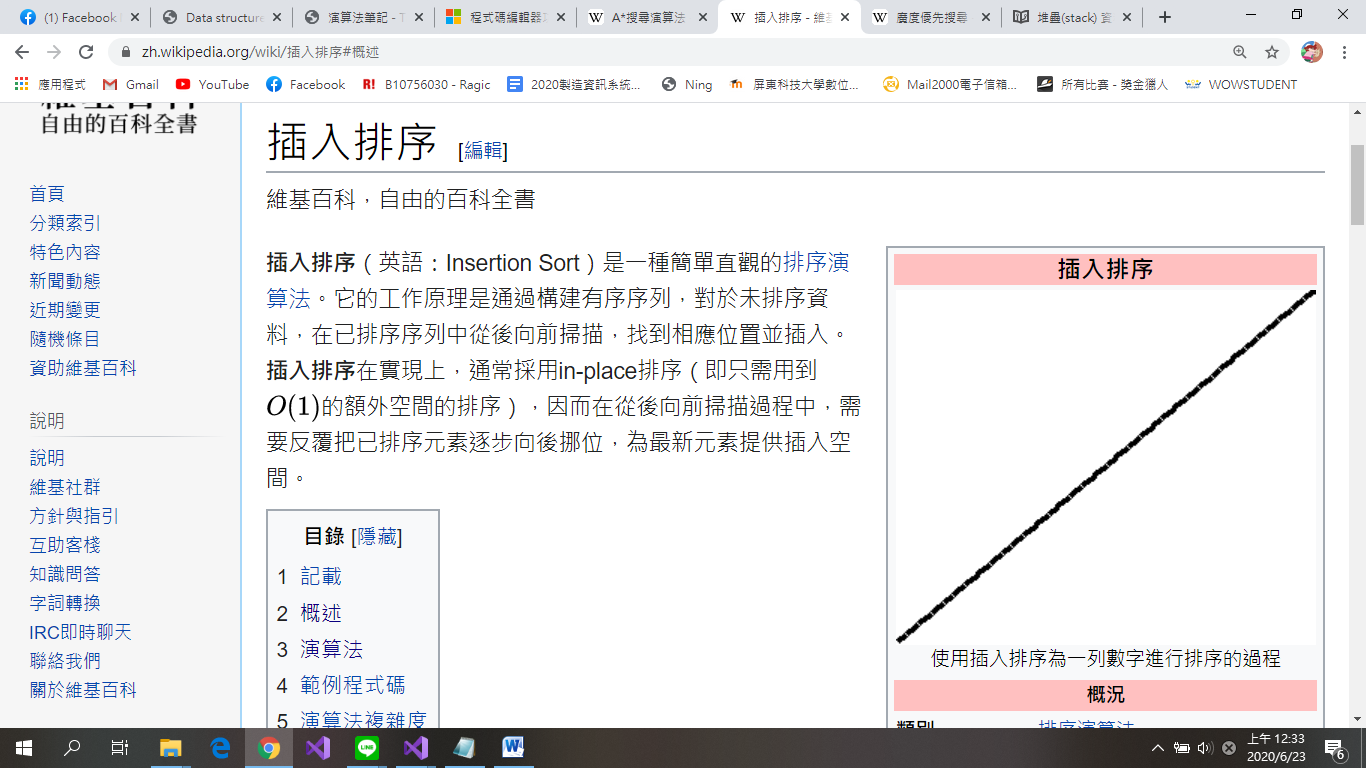
1. **雙向佇列 (Deque)**

　　　一種限定的線型結構，同時具有佇列和堆疊特性的陣列，它可以同時在前端或者後端插入抑或彈出，經常用在有需要變化順序的搜尋或排列，也因此所有的操控僅限於兩端執行。



**圖4.雙向佇列概念圖**

1. **插入排序 (Insertion Sort)**

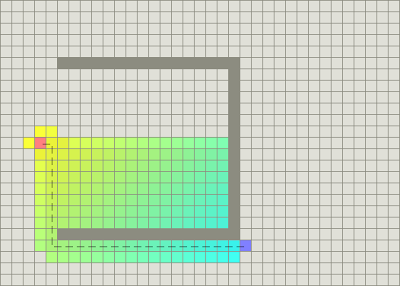
　　　一種排序演算法，通過構建有序序列，對於未排序資料，在已排序序列中從後向前掃描，找到相應位置並插入。因而在掃描過程中，需要反覆把已排序元素逐步向後挪位，如果已排序的元素大於新元素，便將該元素移到下一位置，為最新元素提供插入空位。



**圖5.插入排序概念圖**

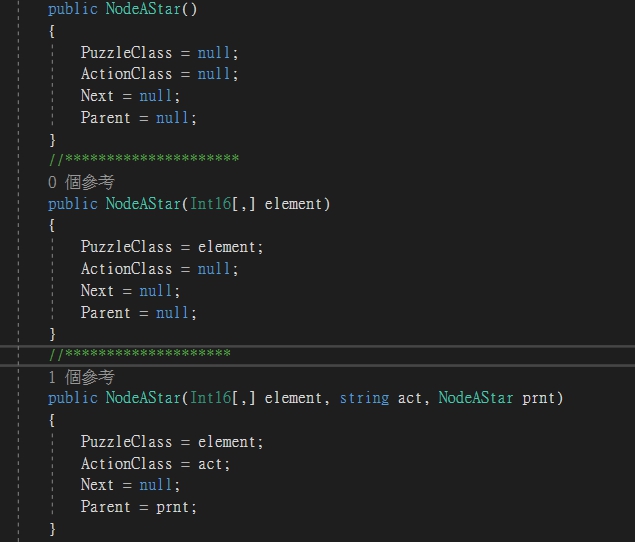
1. **A\*搜尋演算法 (A\* search algorithm)**

　　　一種在圖形平面上，有多個節點的路徑，這個公式避免了一些不必要的路徑排除，在遊戲中通常被應用在尋找最短路徑，所以能用較高的演算效率計算出一條最佳結果。



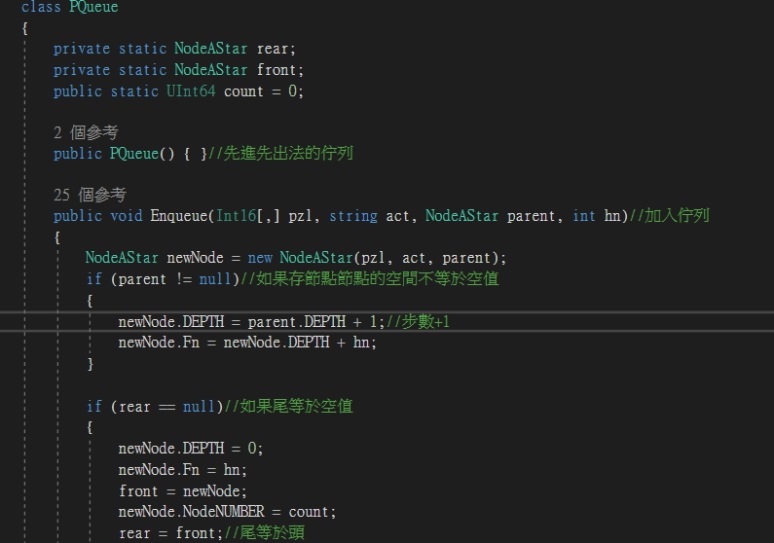
**圖6.A\*搜尋演算法概念圖**

1. **程式碼關鍵部分**
2. **A\*增加節點**

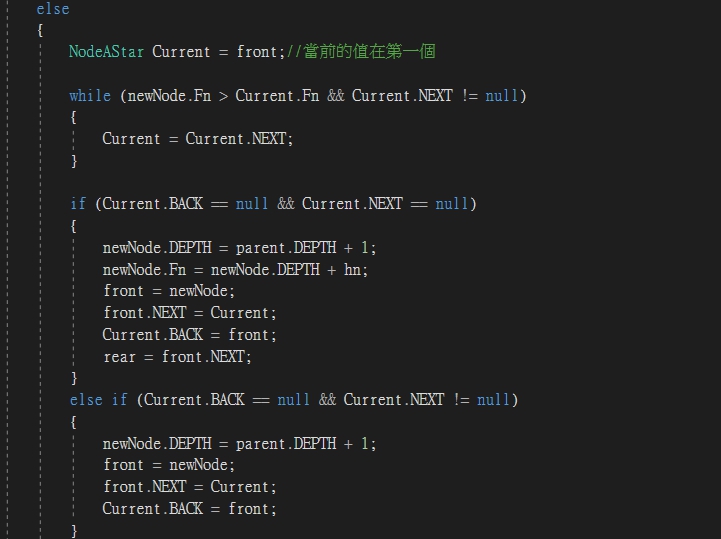
****

**圖7.A\*增加節點程式碼**

1. **A\*佇列**

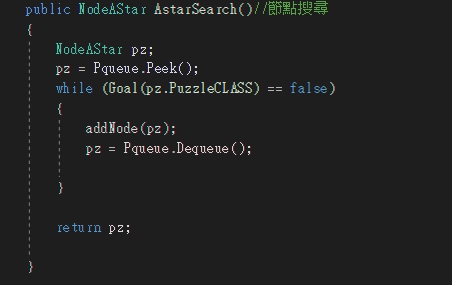


**圖8.A\*佇列程式碼 - 1**



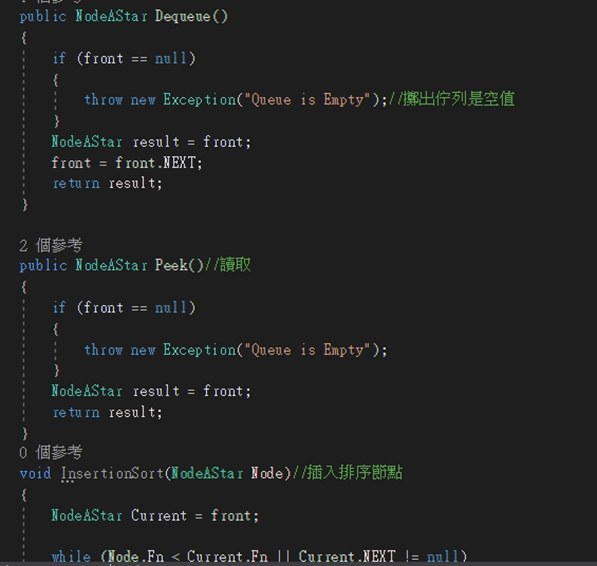
**圖9.A\*佇列程式碼 - 2**

1. **A\*節點搜尋**

****

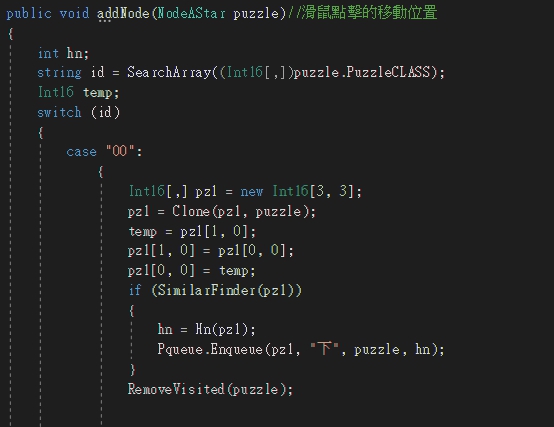
**圖10.A\*節點搜尋程式碼**

1. **A\*雙向佇列**

****

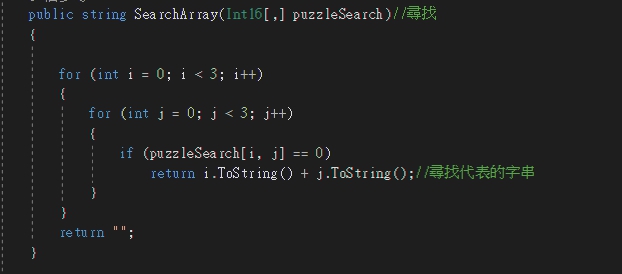
**圖11.A\*雙向佇列程式碼**

1. **A\*滑鼠點擊的移動位置**

****

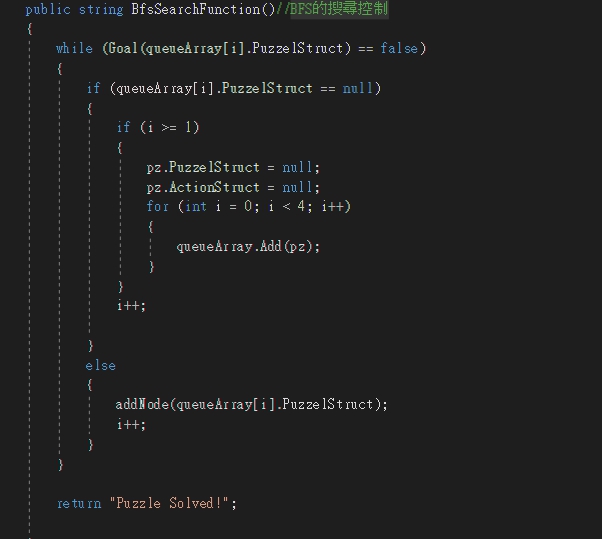
**圖12.A\*滑鼠點擊移動位置程式碼**

1. **A\*尋找代表的字串**

****

**圖13.A\*尋找代表字串程式碼**

1. **BFS的搜尋控制**

****

**圖14.BFS搜尋控制程式碼**

1. **報告中遇到的困難**

* **參考網路資料、解讀與理解**

　　　我們做這項主題時，有參考網路上的資源，但參考的程式碼有些並不能與我們的專案相容，需要做修正，也必須理解程式碼中有哪些是報告會運用到的？這對現階段的我們是一種挑戰。

* **所學的東西應用在報告中**

　　　要把上課所學融會貫通並運用在報告中是一件不容易的事，要想出能用到所學之物的主題，並且實際運用且執行沒有任何問題。

1. **成果展示**
2. **程式設計介面**

　　利用C# Form表單設計一款拼圖遊戲。使用者利用滑鼠點擊表單上的拼圖碎片，將拼圖照阿拉伯數字 1~8排列整齊。

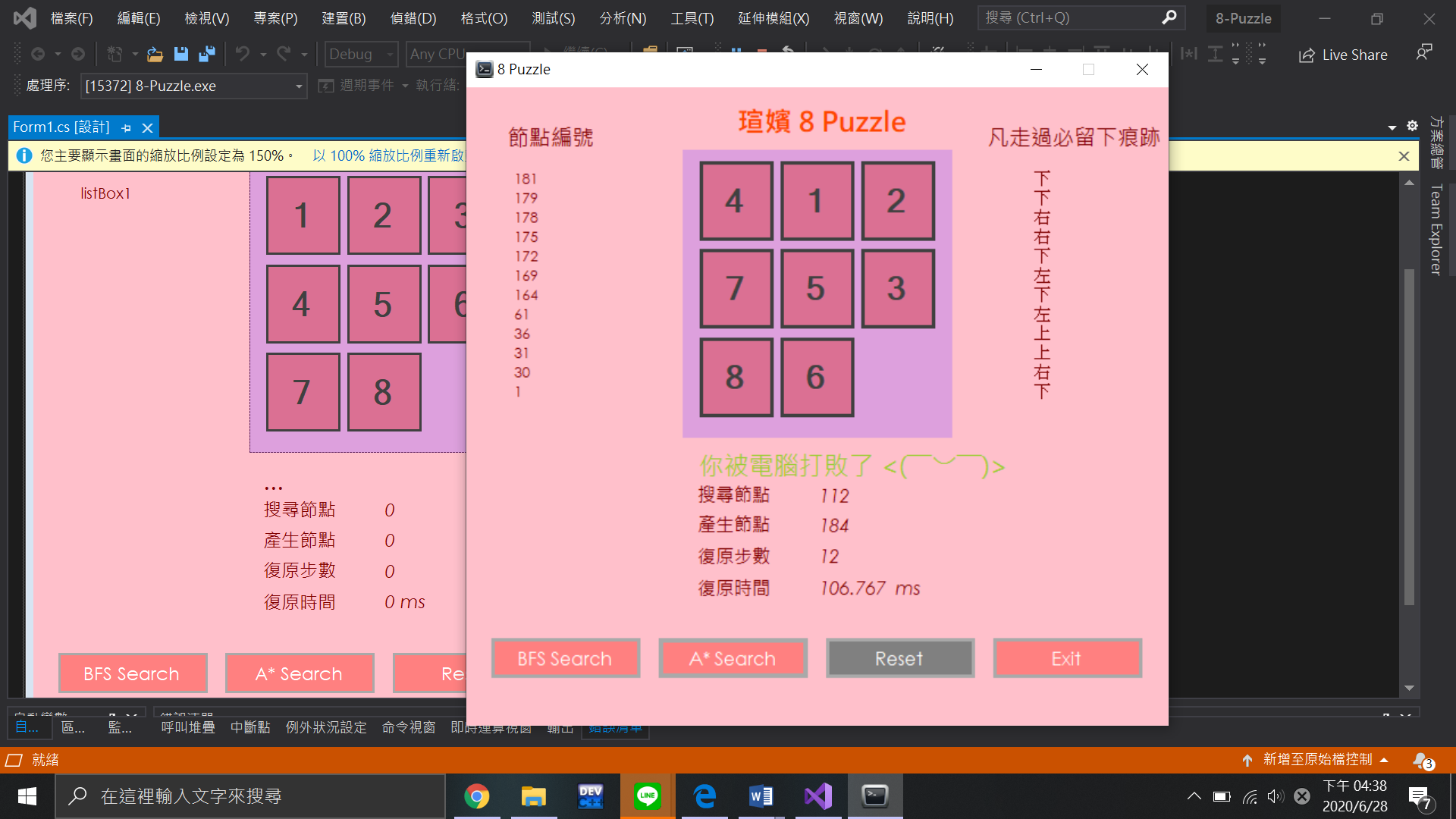
　　若玩家無法完成，可以按下「解答」Button，雜亂的拼圖就會自動搜尋最佳路徑恢復程完整的狀態。



**圖15.成果展示畫面-1**

1. **執行畫面**

　　當遊戲結束，左邊會顯示演算法找出的每個節點編號，而右邊會記錄每一步怎麼移動、怎麼走的，下面會計算總共花了多少時間，走了幾步，搜尋以及產生多少節點。



**圖16.成果展示畫面-2**

1. **小組心得**

**B10756008 王靜文：**

　　這一次的報告是真的有難度，因為要用到這學期所學到的運算法，概念都理解，加上我們選的題目不簡單，實際運用時還是滿困難的。bst或是A\*雖然都是尋找最佳路徑的做法，但實際寫法還是有些差異，這是需要注意的地方。不過在做的過程中，可以了解到所學的東西可以用在哪裡、如何使用，收穫真的不少，也了解到，團隊合作可以讓報告更快更順利的完成。

**B10756015 王滋蘘：**

　　在這次期末報告中，遇到很多挫折，因為要應用到上課所學的東西，很多時候聽懂了理論，但就是不知道程式碼該怎麼寫出來。雖然一開始可能花很久時間，除了要複習這學期的課程，還有小組成員們一起討論研究，這樣東西要怎麼做？演算法要應用在哪裡？在經過一次次嘗試，雖然每次看到bug都很灰心，但最後成果慢慢出來，也讓我們覺得很感動很有成就感。

**B10756027 李昂珊：**

　　透過這次的報告讓我重新審視我這一學期所學到的東西，再加以應用到我們這次的報告中，雖然上課所學到的東西很多，但要轉換成我們想做的報告需要花多一點的時間研究，可是我也就此複習之前上課所學的。團隊中每個人都有他擅長和不擅長的地方，我們互相學習與討論，了解到自己還有哪些不足的地方需要繼續努力學習，我也學習到了團隊合作的重要性。

**B10756030 陳彥芝：**

　　這學期物件導向的課程，運用到了上學期學到的資料結構，如堆疊與佇列、排序演算法及近幾週學習到的樹狀資料結構。這次的期末報告做的是8puzzle，運用到的是樹狀的BFS和A\*樹狀搜尋，運用這兩個演算法找尋拼圖移動的路徑回到最原本的樣子。透過這次的Final project，我們小組更是進一步了解樹狀資料結構，直接用實作方式去理解，使得這門課更有成就感了。

**B10756044 黃珮瑄：**

　　訂題目的時候花了很多時間去討論，一直在想有甚麼東西，能跟這堂課所學的有所結合，最後選擇了大家都有玩過的拼圖小遊戲，它運用之前學到的BFS廣度優先搜尋，去計算玩家移動的路徑，還有A\*搜尋演算法，因為之前沒有實際運用過，所以花比較間去研究，也是把它運用在計算玩家移動的路徑上，在實際的運用過後讓我更了解演算法了。

1. **參考資料**

* 許志仲老師。108下，物件導向程式設計課程。上課教材PPT。
* David（2018.03.21）。C# 用BFS解8-Puzzle問題：<http://davidhsu666.com/archives/csharp-bfs-solve-8-puzzle/>
* [Ajinkya Sonawane](https://blog.goodaudience.com/@sonawaneajinks?source=post_page-----7b509c331288----------------------)（2018.09.16）。Solving 8-Puzzle using A\* Algorithm：

<https://blog.goodaudience.com/solving-8-puzzle-using-a-algorithm-7b509c331288>

* Detlef Beckenbauer。佇列的應用：<https://slidesplayer.com/slide/15138853/>
* 閱讀的程式貓（2020.3.23）。Depth-First Search and Breadth-First Search：<https://dotblogs.com.tw/ilikedogs/2020/03/23/151856>
* Lynn(2017.9.22)。實作Graph與DFS、BFS圖形走訪演算法：<https://kopu.chat/2017/09/22/%E5%AF%A6%E4%BD%9Cgraph%E8%88%87dfs%E3%80%81bfs%E8%B5%B0%E8%A8%AA/>
* kdchang(2016-09-11)。學習資料結構和演算法：<https://blog.kdchang.cc/2016/09/11/javascript-data-structure-algorithm-queue/>
* 維基百科。圖片及補充資料：<https://zh.wikipedia.org/zh-tw/>