小組：27

小組成員：　0515438　　翁邦晏
　　109550059　黃彥傑

AI-G 規劃需求

設計動機

一為 2048 是一個能夠具有明確規範輸贏的遊戲，並且其規則簡單、條目較一般遊戲少，但同時其設計複雜度，以及遊玩時需利用的腦力和遊戲性，也並不輸其他遊戲，因此選定 2048 作為本次遊戲設計作業的主題。

二則為 2048 得具有許多變化性，例如能夠透過增減矩陣階層變換難度，抑或是對戰時能夠使雙方決定對方的下一格產生位置等等，若有足夠的時間得完成此類功能，則能夠使得遊戲更有遊戲性。

另為因應課程需求，需利用 LabView 並且與組員組成一組，共同完成一個遊戲並設計出其 AI，考量本組組員雙方能力要製作 AI 稍有難度，故選定此遊戲，一方面網路資源足夠充分做為參考，另一方面也能更進一步打磨 LabView 的基礎。

需求分析

首先必須先建構出一個 4 x 4 的方格作為 Front Panel 的呈現，此專案中，將利用二維矩陣的特性，以及 Property Node 中的 Value 完成。

再者，需要隨機在未被占用的格子內隨機產生 2 或 4。

三為隨著方格內數值的變化，需要改變格子的顏色，以利玩家對於其不同的分數做辨識。此專案中，以將數值對應的顏色先做定義，再判定二維陣列中的數值，對格子進行變色。

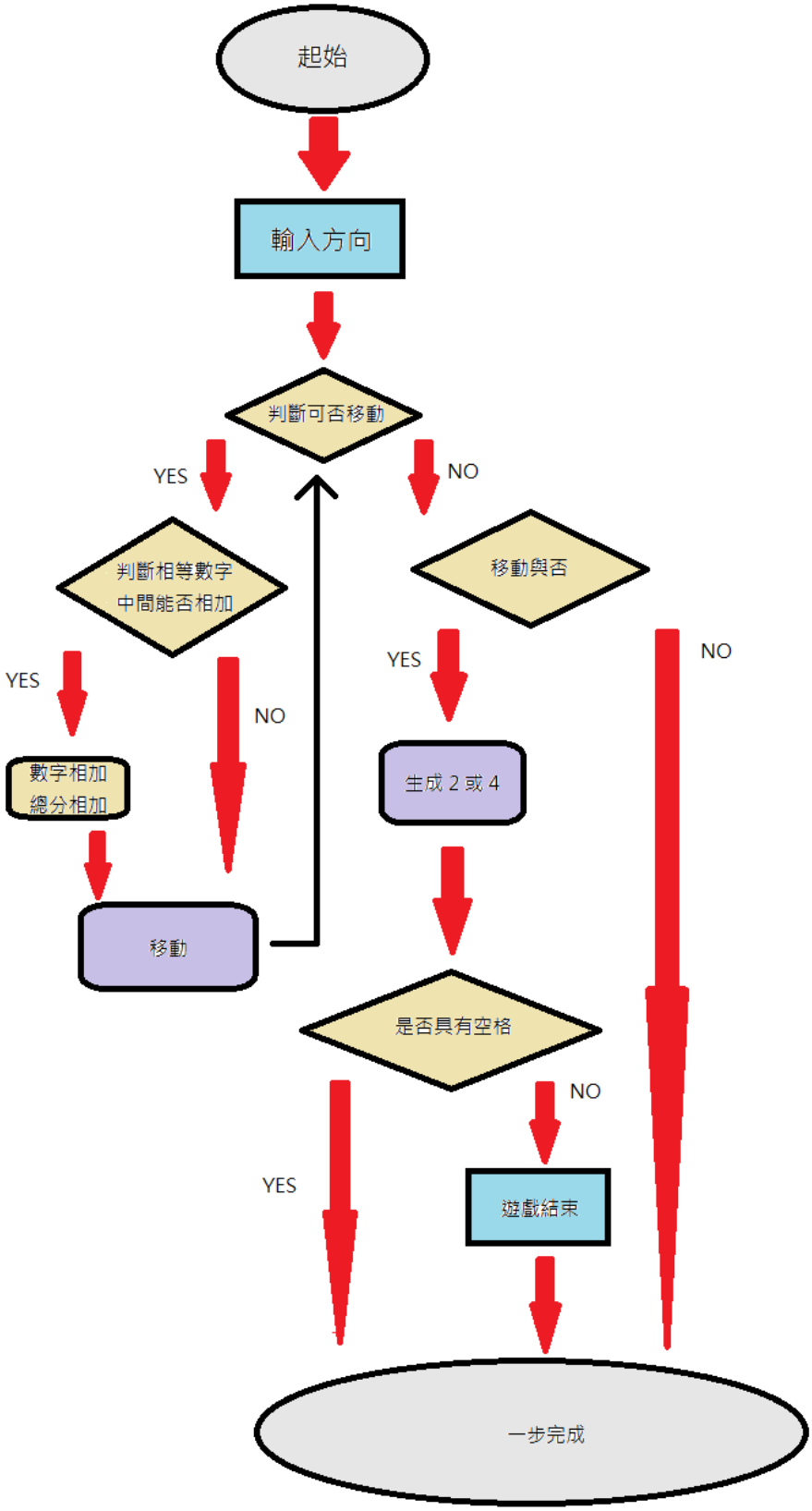
四為分數加總。在判斷格子合成的同時，能夠完成此操作。

五則是移動，也是本遊戲的核心。這邊將使用 Python 程式碼完成。

六，顯示對戰兩方何者勝利。該功能將採用動態判斷，以一邊進行遊戲，一邊比對分數，並顯示兩方何者勝利。

其餘還有其他次要功能如下：遊戲時播放背景音樂、連線對戰功能以及調整 AI 對手難度。

程式規劃流程


起始
輸入方向
判斷可否移動
YES
NO
判斷相等數字
中間能否相加
移動與否
YES
NO
YES
NO
數字相加
總分相加
生成 2 或 4
移動
是否具有空格
NO
YES
遊戲結束
一步完成

AI-G 規則說明

對弈方式

單人模式：

單人模式底下，一開始會於隨機兩格中產生２或４，透過上下左右移動對相同數值的數進行合成，其後每移動一步，都會隨機產生一格２或４以供合成。

最終目標為達成４ｘ４的方格中，最大數值達到 2048 或以上，並且針對分數進行相加，每當合成一個數值，分數將會加上其數值大小，舉例來說：２和２合成後，總分會加上４。

當玩家所有的格子不能再被合成後，遊戲結束。

雙人模式：

雙人模式底下，基本運作與單人模式相同，兩方皆會產生隨機產生２或４，以最終分數進行輸贏比較，意即當其中一方已經遊戲結束，而另一方遊戲尚可進行時，遊戲並不會終止，將以兩方最終分數進行比較。

分數核計

分數核計方式如單人模式所述，相同數字加總後，會於總分上加上新合成的數值，若同時合成了兩個數字，像是同時合成 4 和 4096，會在總分上同時加上此二數字，也就是一步內會加上 4100。當無法再進行合成時，總分即最終分數。

對弈結果判定

對弈的最終結果，以單人模式來說，即以玩家是否能夠進行下一步進行判定，若玩家最終無法進行下一步合成，則遊戲結束，並得到最終分數。

以雙人模式來說，會以雙方玩家是否能繼續進行遊戲做判定，當雙方玩家皆結束遊戲，則比較最終分數，最終分數勝者贏得比賽。此間，若其中一名玩家已經結束遊戲，並且分數低於另一尚可遊玩玩家，則玩家得自行決定是否繼續遊戲。

AI-G 玩家說明

AI 玩家設計理念

從 2048 的規則出發，其規則很明確，僅有的四個方向中數字相同並且中間沒有其他相異數字的兩格，能相加獲得分數，若是無法再進行移動則遊戲結束，以最終成績作為評判標準。由此可知，我們得透過模擬往後移動的方式所得到的分數，尋找出最適當的移動方式。

另於 2048 遊戲的 AI 中，模擬餘下步驟的方式，尚有幾個可以考量的問題，最基本的，目前所擁有的格子及其數值，接著是剩下的空格數。目前的格子數能直接影響下一步的分數加總，空格數則決定還能有多少空間移動，基本上越少

空間移動，對於玩家來說也就更困難。再者是遞增性，舉例來說，倘若能做成一條 16 => 8 => 4 => 2 的連接排列，則相對的有機會得到最大值。
第一段的方法實踐於本次所有 AI 設計中，第二段的方法，則僅使用於 Expert。

預計導入的方法

Idiot、Normal：

此處將採用蒙地卡羅方法，對餘下步驟模擬移動，並以模擬得到最高分數採取最終行動。
Idiot 的部分，設計方面將後續模擬步數設置為 1 步，模擬次數為每個方向移動 3 次，以模擬後所得到的最高分數為下一步移動的基準。
Normal 部分則將後續模擬步數設置為 30 步，總模擬次數為每個方向 40 次，依照最終計算所得出的最大值成為下一步移動的基準。
由兩者的次數可以看出，很明顯的 Normal 的 AI 會比 Idiot 更加精準，雖然不可避免 Idiot 可能在機率極小，近乎不可能的情況下，每一步都正好是最佳解而造成可能勝過 Normal 的情況，但可以確定在常理之下，Normal 所得到的解，最終一定優於 Idiot。
實際測試時 Normal 可以保證到達 2048，也就是遊戲目標，而 Idiot 大多僅能合成 256。

Expert：

此處原先應該由本組自行設計，然本組組員對於程式設計以及 AI 的相關知識尚淺，無法自行設計出相應難度的 AI，故為了遊戲性以及對戰性，直接導入網路上他人的程式碼。該程式碼經過測試得到達 16384，用以做為高手的對戰對手增加其遊戲的挑戰性。

參考資料

nneonneo(2015) 2048-ai. 2048-ai/2048.cpp at master · nneonneo/2048-ai · GitHub
Gabriel Romualdo(2020) Monte Carlo Tree Search Algorithm for 2048. Using the Monte Carlo Tree Search Algorithm in an AI to Beat 2048 (and other games)
MikhailLenko(2020) build-2048-in-python. python-youtube-code/build-2048-in-python at master · kiteco/python-youtube-code