

AI 作業(一)：鬼的隨便 AI

2015 資訊之芽語法班 suhorng

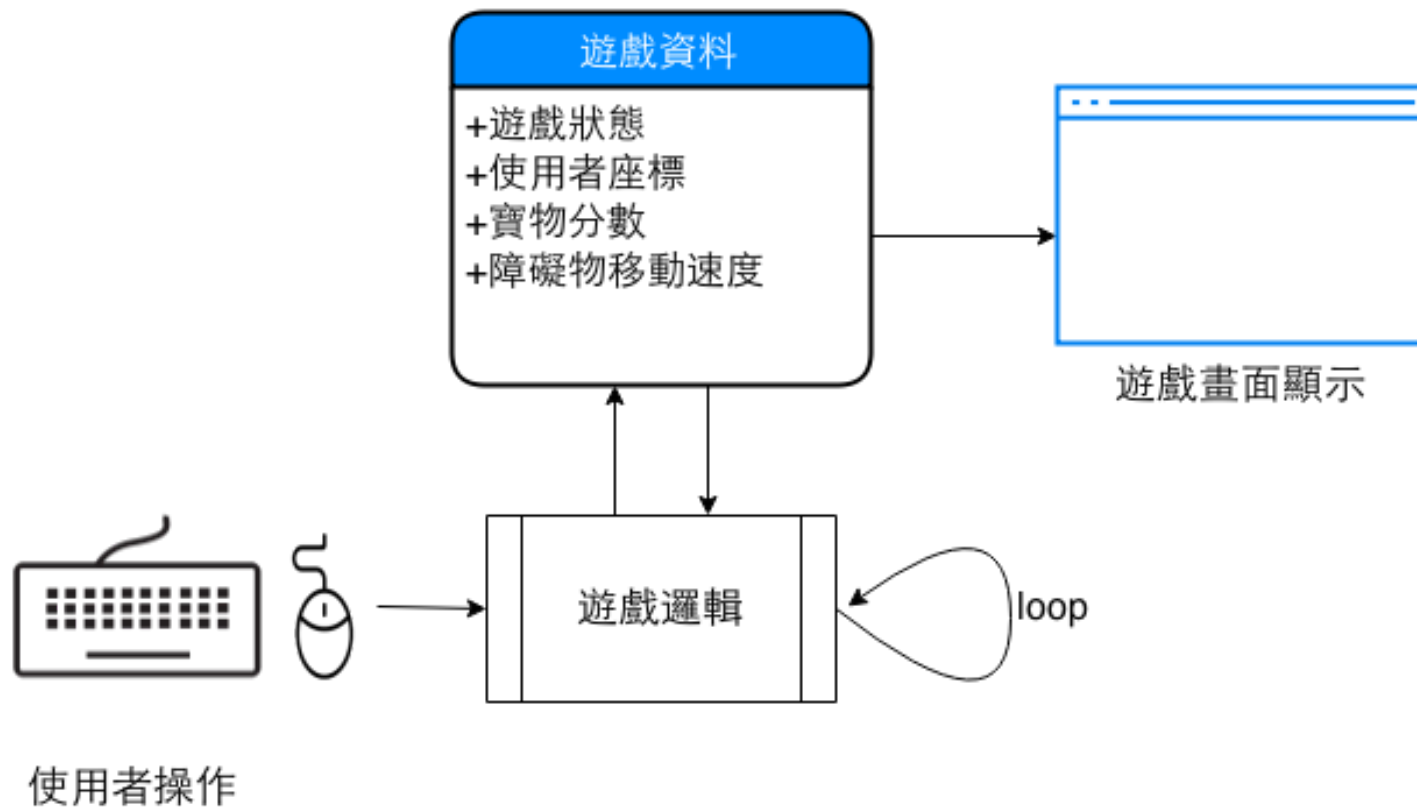
小遊戲 AI 作業

- 鬼的(基本款) AI (本週)
- 人的 AI (下週, 下下週)

小遊戲複習

1. 遊戲邏輯
2. 遊戲資料
3. 使用者操作
4. 畫面生成
5. 資源控管

小遊戲概念 (一般/粗略而言)

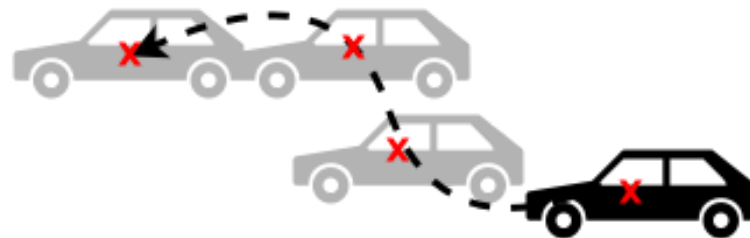


小遊戲概念: 遊戲邏輯

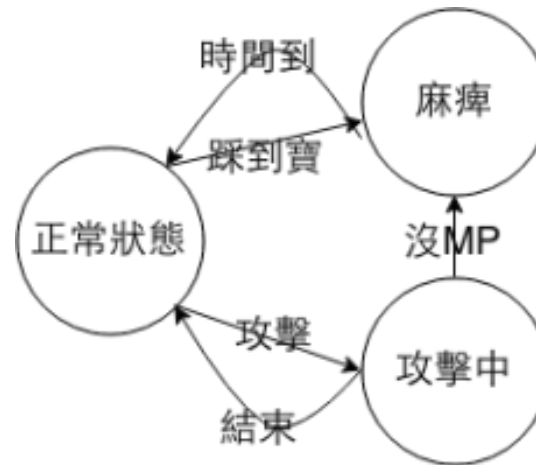
- 雖然概念上可能是連續的, 但...



- 取樣點跟電腦的計算是離散的



遊戲邏輯相關知識: 自動機 & 狀態



```
const int ST_NORMAL = 0, ST_PARALYZED = 1, ST_ATTACKING = 2;
int user_state = ST_NORMAL, user_time = 0;

// 程式迴圈
while (...) {
    if (user_state == ST_NORMAL) { // 處理輸入 ⇒ 移動
        if (IsInBox(x, y, block_x, block_y)) user_state = ST_PARALYZED;
    } else if (user_state == ST_PARALYZED) {
        if (user_time > 10) user_state = ST_NORMAL; // 結束麻痺
        else
            ++user_time;
    } else if (user_state == ST_ATTACKING) { // 處理敵人生命值
        if (user_mp < 0) user_state = ST_PARALYZED;
    }
}
```

AI 要實作的函式

- 拉到 `ai.cpp` 檔案最下面

```
// 對每隻鬼，每個回合會被呼叫
// 如果回傳的不是合法的方向(例如有障礙物)，則會被忽略掉

// 參數：
// - i: 代表目前處理的是第 i 隻鬼( ghost[i] )
// - 回傳值：鬼的下一個方向，DIR_UP, DIR_DOWN, DIR_LEFT,
//           DIR_RIGHT 或 DIR_NONE
int GhostDecide(int i) {
    ...
}
```

假如 `ghost[i]` 與玩家的曼哈頓距離小於 10, 則從上下左右可以走的地方中, 選擇走了之後與玩家曼哈頓距離最近的

若與玩家的曼哈頓距離大於等於 10, 則 random 挑一個可以走的地方或不動

AI 要實作的函式

- 兩個點 (x_1, y_1) 和 (x_2, y_2) 的曼哈頓距離為 $|x_1 - x_2| + |y_1 - y_2|$ (座標差的絕對值的和)
- 所以要寫的 AI 大致思路是

若 曼哈頓距離(ghost[i], 玩家) < 10:

挑出一個方向 {上,下,左,右}, 使得

* 這個方向可以走 (不是牆壁, OBJ_WALL)

* 走了之後, 與玩家的曼哈頓距離最短

假如有很多個方向符合, 隨便哪個都可以

否則:

從 {不動 or 上下走右可以走的方向} 中隨機挑一個方向

- 回傳常數 DIR_UP, DIR_DOWN, DIR_LEFT, DIR_RIGHT, DIR_NONE 之一

遊戲資料

- 玩家位置: `pacman.x` 跟 `pacman.y`
 - 玩家狀態: 生 `PACMAN_ALIVE`, 等復活 `PACMAN_DEAD`
 - 碰到鬼會死一段時間後重生
- 鬼 `ghost[i]` 的位置: `ghost[i].x`, `ghost[i].y` ($0 \leq i < 3$)
- 遊戲地圖: `map[lx][ly]` ($0 \leq lx < \text{WIDTH}$, $0 \leq ly < \text{HEIGHT}$)
 - 牆壁: `map[lx][ly] == OBJ_WALL`
 - 金幣: `map[lx][ly] == OBJ_GOLD`
 - 沒東西: `map[lx][ly] == OBJ_NONE`

遊戲邏輯

- 固定時間會呼叫 `SystemTimer(int)`, 裡面會呼叫 `PacmanUpdate()` 和 `GhostsUpdate()`

```
void GhostsUpdate() {  
    ....  
    int nxt_dir = GhostDecide(i);  
    ....  
}  
  
void PacmanUpdate() {  
    ...  
    int nxt_dir = PacmanDecide();  
    ...  
}
```

- `GhostDecide(i)` 應該要回傳 `ghost[i]` 下一個回合要走的方向, `PacmanDecide()` 同理. 本週作業是寫 `int GhostDecide(int i);`, 範本提供在檔案最後面了