



棋盤回合制 遊戲

生工二戴琮愷b11602054
經濟一黃昱祺b11303006
農藝二許懷莊b11601032
公衛二卓庭榛b11801004

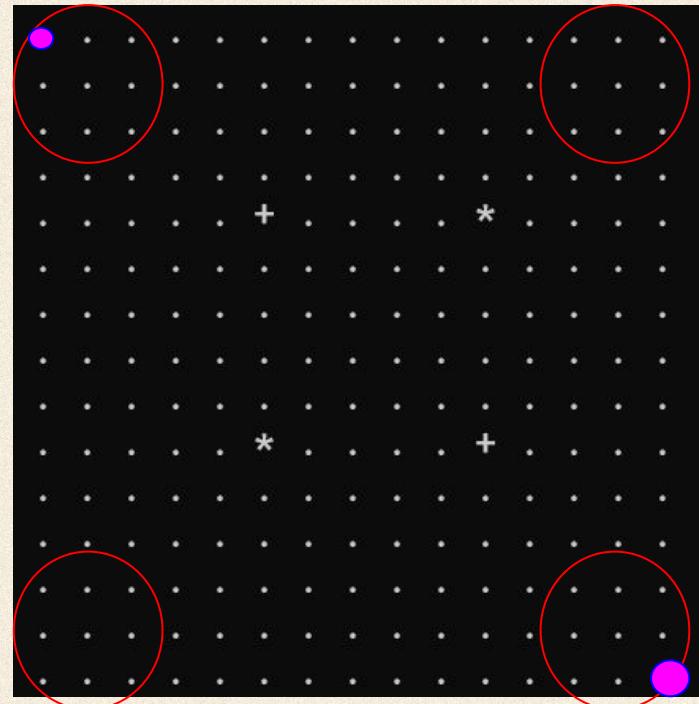


基本描述

- 遊戲模式：棋盤回合制
- 隊伍組成：兩隊，每隊三個角色，由兩名玩家操控，一方操作完剩餘角色，才換另一方，後手將會以小寫字母表示
- 玩法：每一輪由一隊操縱其所有當前存活之角色，其可選擇“攻擊”或“移動”玩家可藉由地圖佔領、搶奪物件及角色特性設法將另一方所有角色擊潰
- 勝利條件：消滅對方所有角色

初始化

- 棋盤為 15×15 , 角色初始擺放位置只能在
角落 (我們有用try-catch擋住不合理的位置)
- 位置表示 : (x,y) , (橫向、縱向)
- 紫色點的位置是 $(0,0)$, 邊界是 $(14,14)$, 向
右為正, 向左為負, **向下為正**, 向上為負



角色選擇

- 使用者會根據prompt輸入角色職業，職業後之數字表第幾隊
- 輸入此角色放置數量，同一角色最多放置2隻並緊接輸入想放置之座標，座標規定為x,y皆在0~2或12~14間
- 另一隊亦同上步驟輸入

```
Which character do you want to add in team1? ( Wizard1, Knight1, Archer1, Tank1 )
Wizard1
You can at most add 3 characters.
Quantity: 2
Location: 2 2
Location: 12 12
Which character do you want to add in team1? ( Wizard1, Knight1, Archer1, Tank1 )
Knight
Error: Invalid character type. Please try again.
Knight1
You can at most add 1 characters.
Quantity: 1
Location: 2 12
You cannot use team1's members' locations!!!!!!
Which character do you want to add in team2? ( Wizard2, Knight2, Archer2, Tank2 )
```

各角色數值呈現

Wizard:

```
member_number 0,Name: Wizard1  
HP: 80/80  
Attack: 100
```

Tank:

```
member_number 0,Name: Tank2  
HP: 220/220  
Attack: 45
```

Archer:

```
member_number 0,Name: Archer1  
HP: 80/80  
Attack: 100
```

Knight:

```
member_number 1,Name: Knight2  
HP: 180/180  
Attack: 60
```

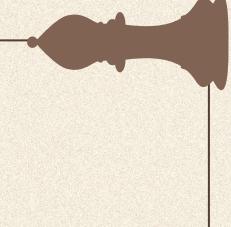
角色攻擊範圍

1.Wizard : 4 X 4 的方格內之“未死亡敵方隊伍角色”

2.Knight : 攻擊範圍為以自身為中心之十字，十字大小分別為上下左右各 2 格內之“未死亡敵方隊伍 角色”

3.Tank : 3 X 3 的方格內之“未死亡敵方隊伍角色”

4.Archer : 4 X 4 的方格內之“未死亡敵方隊伍角色”



角色移動範圍

- 為了防止角色能任意移動至想移動的地方，我們替各個角色的移動添加了限制：

Wizard : 橫向移動之格數 + 縱向移動之格數 6 格內(包含)

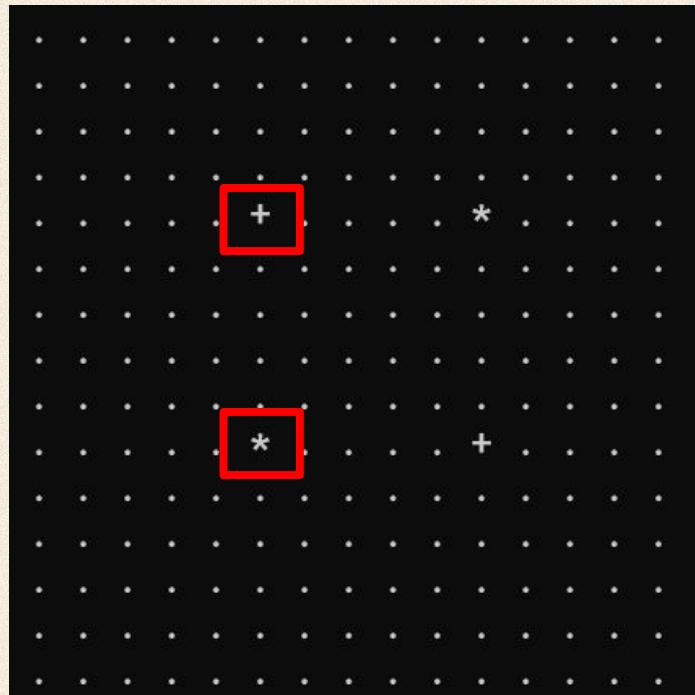
Knight : 橫向移動之格數 + 縱向移動之格數 5 格內(包含)

Tank : 橫向移動之格數 + 縱向移動之格數 4 格內(包含)

Archer : 橫向移動之格數 + 縱向移動之格數 6 格內(包含)

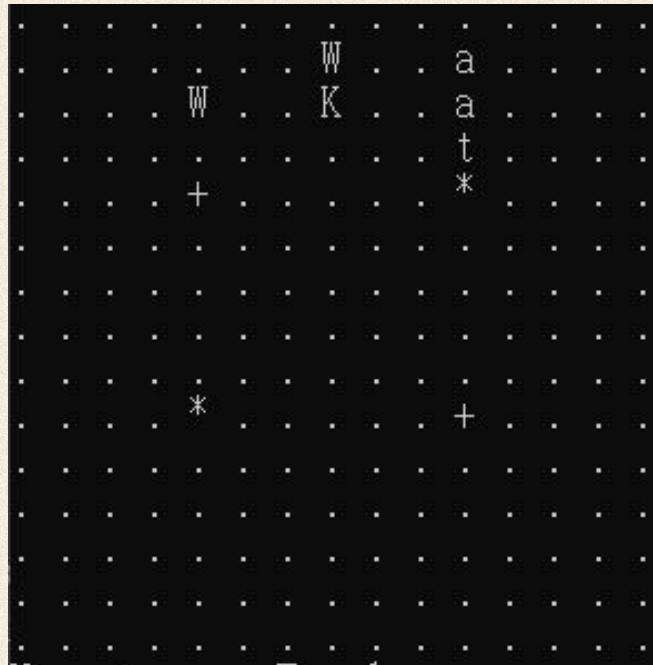
特殊道具

- “+”: 當有任何角色走到這一格內，即獲取此道具，並且道具消失，回補 30 點血量
- “*”: 當有任何角色走到這一格內，即獲取此道具，並且道具消失，獲得 20 點的攻擊力加。



遊戲Tips

- 道具的加成效果非常重要，可作為戰術上的誘餌，也可能延伸出圍繞著道具的防禦佔位
- 保護攻擊主力 Archer 跟 Wizard
- Tank 佔住道具周圍重點格子或搶走道具
- Knight 試探、破壞對方陣型
- 搶到攻擊力道具的 Knight 會變成遊戲中強大角色棋子





遊戲心得與未來展望

- Team的部分牽扯到非常多指標的操作，很容易出差錯，尤其是在後期800多行以後，一個小錯誤常會花上1~2個小時處理
 - 另外因為我們做的是棋盤遊戲，我們一方面處理道具以及角色的攻擊以及觸發效果的邏輯，還要考慮到螢幕顯示上的問題。
 - 我們投入了大量時間和精力來處理上述的技術細節與挑戰，這影響了我們考慮遊戲平衡性的時間。
- 
- 