

Assign 4

Array

Requirements

- ★ 利用int[][] soilHealth管理各格土壤的生命值，沒有石頭、一層石頭和兩層石頭的土壤預設生命值分別為15/30/45，並用此陣列數值顯示對應土壤與石頭圖片；各區土壤層石頭排列比照作業三規定。
- ★除了第一層外，其餘每層土壤隨機 1 ~ 2 格挖空，並在挖空的格子顯示土壤背景圖片(soilEmpty.png)。每次開始遊戲需重新隨機！
- ★ 土撥鼠移動結束到達定點時，如果下方沒有土壤則受重力影響強制往下移動。
- ★★ 每四層為一區（總共六區），在每一區隨機選一格放入蔬菜。利用float[] cabbageX, cabbageY管理蔬菜位置（陣列大小為6）；每次開始遊戲需重新隨機；碰撞判定參考作業二規定，而當土撥鼠擁有最大生命值(PLOYER_MAX_HEALTH)時碰到蔬菜則沒有反應。生命值顯示參考作業三規定。
- ★★ 每四層為一區（總共六區），在每一區土壤中隨機挑一層放入一個士兵，並分別選 X 軸範圍內任一位置作為移動的起點。利用float[] soldierX, soldierY管理士兵位置；每次開始遊戲需重新隨機；移動方式與碰撞判定參考作業二規定；土撥鼠碰到士兵回到地表時，填回初始位置下方的土壤。

Requirements

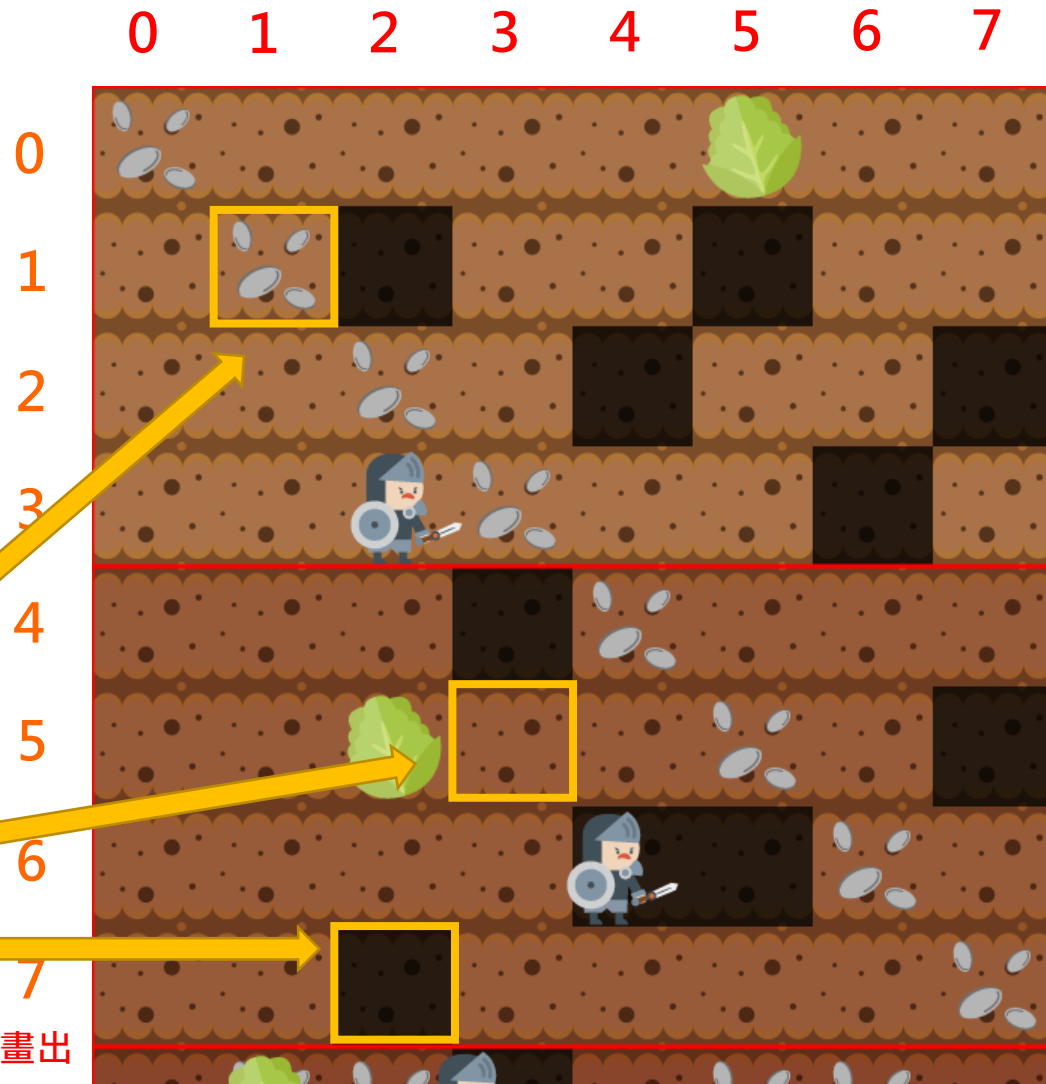
★★★ 實作土撥鼠挖掘：

- 土撥鼠只能在沒有土壤（生命值為零）的格子移動。
- 當土撥鼠往有土壤的格子移動時，**按住方向鍵**會停留在原地挖掘該格土壤、扣除土壤生命值。
- 挖掘時須隨挖掘方向改變土撥鼠圖片，**而過程中放開方向鍵即可隨時回到閒置狀態**。
- 沒有石頭、一層石頭和兩層石頭的土壤分別需要挖掘 15/30/45 畫格 (0.25/0.5/0.75秒)。
- 需依照土壤生命值改變該格土壤與石頭的顯示圖片（見投影片p.7-9）
 - 將需要用到的圖片讀取至PImage[][] soils, stones中

```
int[][] soilHealth
```

`soilHealth[欄][列]`

```
例： soilHealth[1][1] = 30
      soilHealth[3][5] = 15
      soilHealth[2][7] = 0
```



紅線標示不須在作業中畫出

隨機空格

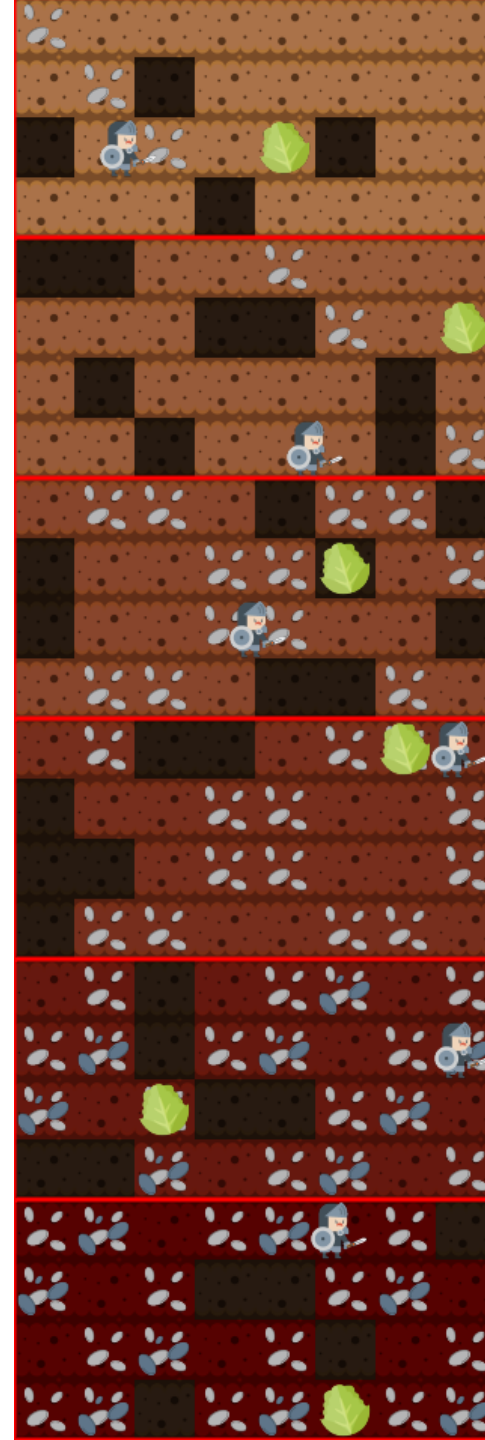
注意：
為避免土撥鼠一開始遊戲就往下掉，
第一排不須挖空！

隨機決定每一層空格格數 (1 ~ 2)



隨機決定各層要挖空的格子

紅線標示不須在作業中畫出

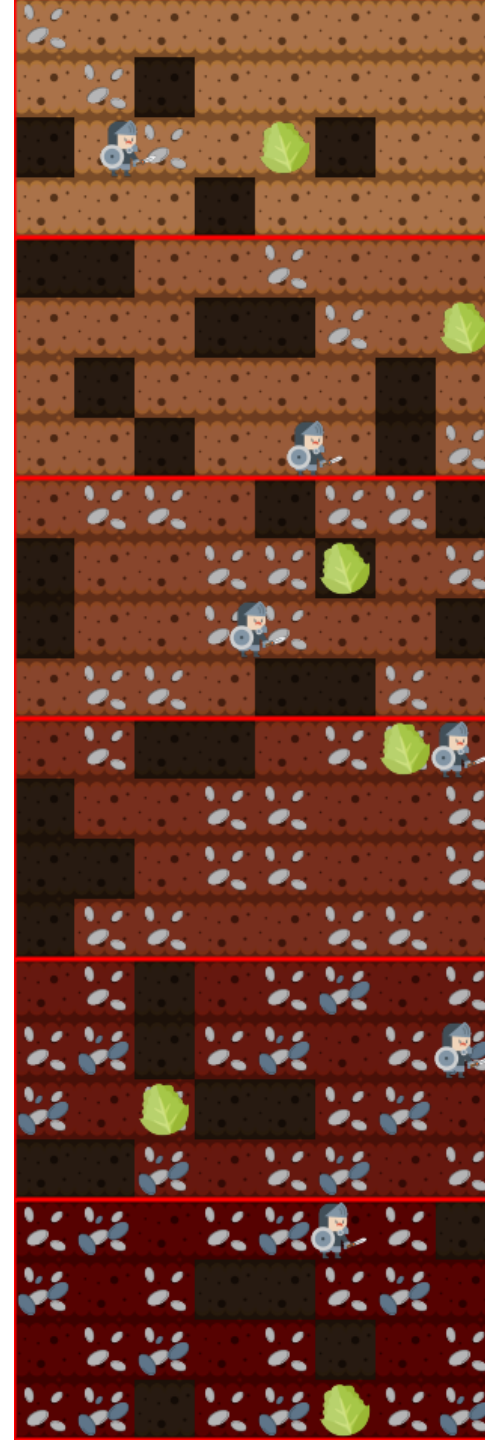


士兵/蔬菜

每四層為一區（即按照土壤顏色分區）

每區隨機挑選一格放一個蔬菜

每區隨機挑選一層放一個士兵，起始 X 位置分別隨機



紅線標示不須在作業中畫出

土壤圖片路徑

提示：因圖片數量多，請在setup中利用for迴圈讀取各個圖片進入陣列

空格用的土壤背景圖片：
img/soils/soilEmpty.png
(所有深度的空格皆使用此圖)



```
PImage[][] soils = new PImage[6][5];
```

```
soils[種類][狀態]  
= loadImage("img/soils/soil種類/soil種類_狀態.png");
```

```
例：soils[3][1] = loadImage("img/soils/soil3/soil3_1.png");
```

種類	狀態	0	1	2	3	4
	對應生命	1 - 3	4 - 6	7 - 9	10 - 12	13 - 45
0						
1						
2						
3						
4						
5						

石頭圖片路徑

提示：因圖片數量多，請在setup中利用for迴圈讀取各個圖片進入陣列

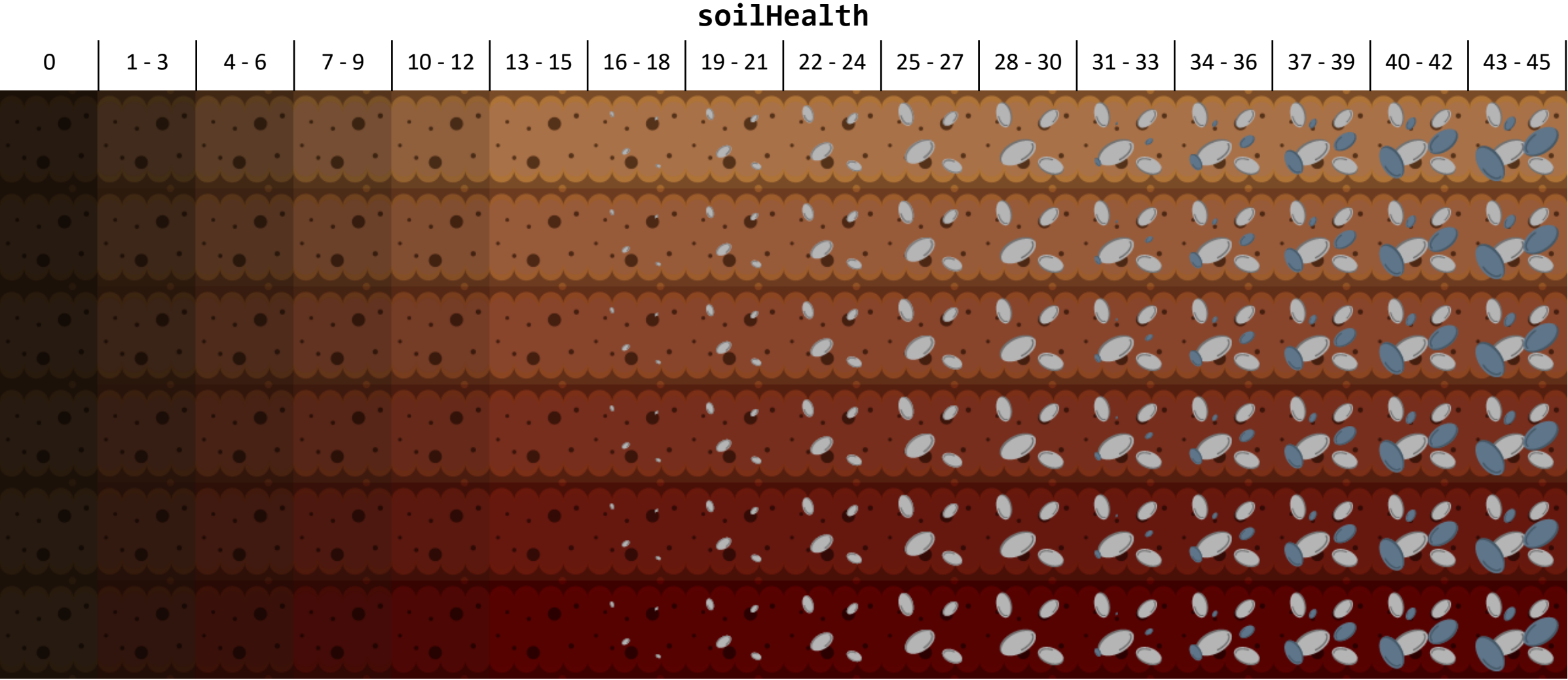
```
PImage[][] stones = new PImage[2][5];
```

```
stones[種類][狀態]  
= loadImage("img/stones/stone種類/stone種類_狀態.png");
```

```
例：stones[1][1]  
= loadImage("img/stones/stone1/stone1_1.png");
```

種類 \ 狀態		0	1	2	3	4
0	對應生命	16 - 18	19 - 21	22 - 24	25 - 27	28 - 45
	圖片					
1	對應生命	31 - 33	34 - 36	37 - 39	40 - 42	43 - 45
	圖片					

土壤圖片隨土壤生命值變化



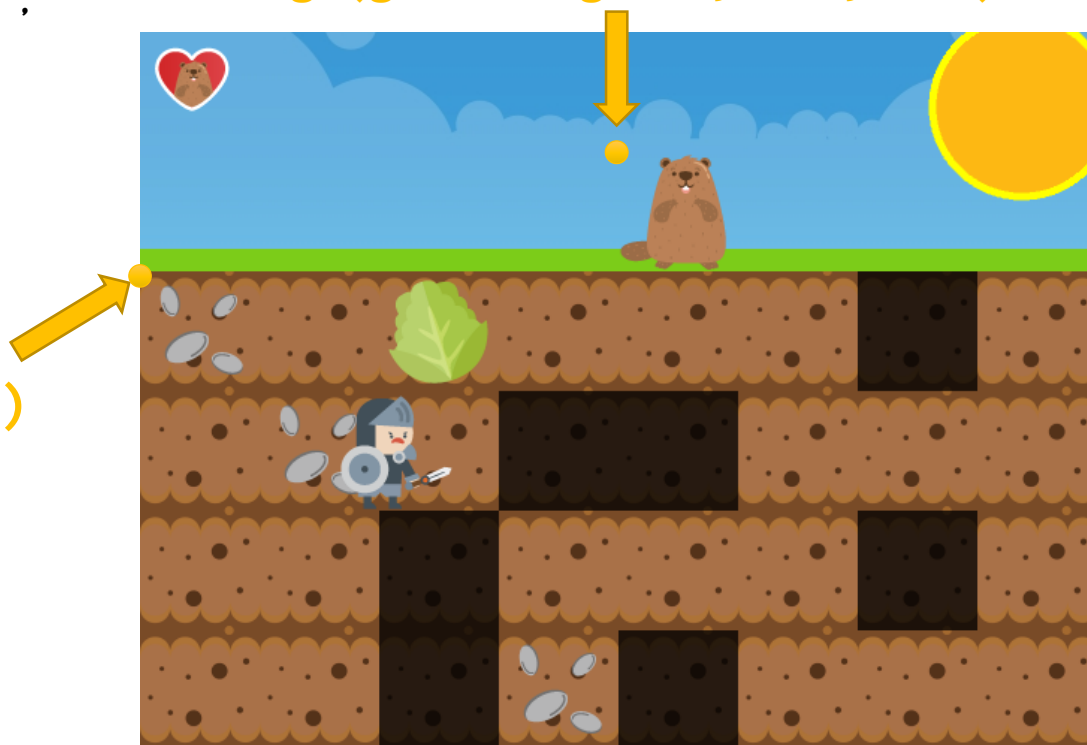
預設程式碼

本次作業預設程式碼提供基本場景配置、流暢移動以及視角跟隨的程式碼，
並提供許多作業提示可供參考，
同學亦可以選擇使用自己先前的程式碼來修改，但請務必確保能正常運作。

為了方便計算土壤與土撥鼠位置、不需要每次畫圖都扣掉地面兩層，
預設程式碼利用translate將地表設為 $Y = 0$
如果使用預設程式碼，
須注意(0, 0)為第一層土壤的左上角位置
而土撥鼠起始點為(320, -80)

`image(soil0, 0, 0)`

`image(groundhogIdle, 320, -80)`



Demo用程式碼

預設程式碼包含demo影片中的數字顯示功能，按下 **B** 鍵可以切換顯示 soilHealth 數字在對應的土壤格子上



Demo 影片

<https://www.youtube.com/watch?v=Rq27zcNMyO4>