Assign 5

Function

Requirements

★ #1 加入時鐘道具，生成的位置條件與蔬菜相同，但不可與蔬菜共用一格；參考 void initCabbages() 做法，實作 void initClocks() 並在 void initGame() 中呼叫使用。

★ #2 設計一個 void addTime(float seconds) 功能，當土撥鼠碰到時鐘時，時鐘消失並呼叫此功能增加 15 秒的剩餘時間。

★ #3 參考Axis-Aligned Bounding Box(AABB)碰撞偵測方式設計一個boolean isHit(float ax, float ay, float aw, float ah, float bx, float by, float bw, float bh) 函式，並利用此功能來判斷『土撥鼠是否吃到蔬菜/時鐘』與『敵人是否撞到土撥鼠』。

★★ #4 修改 void drawTimerUI() 文字顯示內容，設計一個 String convertFramesToTimeString(int frames)將畫格數轉為 mm:ss 字串。

e.g. 剩餘 3906 frames = 65.1s = "01:05" (採無條件捨去至秒數位)

Requirements

★★ #5 修改 void drawTimerUI() 文字顯示顏色，設計 color getTimeTextColor(int frames)，並利用此功能在不同剩餘時間取得對應的顏色：

大於等於兩分鐘：天藍色(#00ffff)

小於兩分鐘、大於等於一分鐘：白色(#ffffff)

小於一分鐘、大於等於三十秒：黃色(#ffcc00)

小於三十秒、大於十秒：橘色(#ff6600)

小於十秒：紅色(#ff0000)

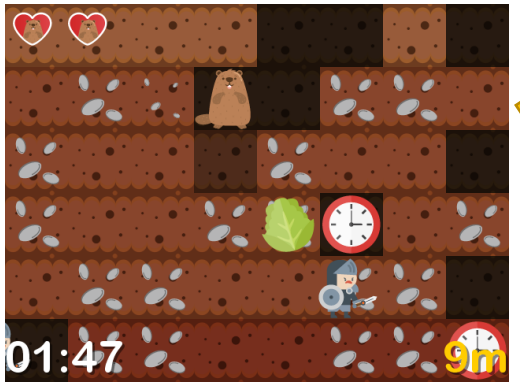
★★★ #6 實作判斷任一層士兵位置的函式 int getEnemyIndexByRow(int row)，如該層有士兵則傳回該士兵在士兵陣列中的索引值(i)，並實作 void drawCaution() 在該士兵上方顯示警示圖片；如該層無士兵則傳回-1，不需顯示圖片。

遊戲流程 (於初始程式碼已完成)

GAME_START

gameTimer = GAME_INIT_TIMER

GAME_RUN



運行過程中
每次draw()都會將gameTimer減一

土撥鼠成功回家

GAME_WIN

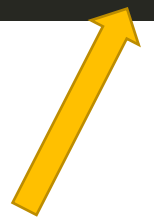
gameTimer 或 playerHealth <= 0

GAME_OVER

gameTimer

已宣告int gameTimer計時器與相關常數，請直接使用

```
final int GAME_INIT_TIMER = 7200;
int gameTimer = GAME_INIT_TIMER;

final float CLOCK_BONUS_SECONDS = 15f;
```



註：15f 即 (float) 15，
Processing不需輸入f即會自動轉型為float

```
void keyPressed(){
    if(key==CODED){
        switch(keyCode){
            case LEFT:
            leftState = true;
            break;
            case RIGHT:
            rightState = true;
            break;
            case DOWN:
            downState = true;
            break;
        }
    }else{
        if(key=='t'){
            gameTimer -= 180;
        }
    }
}
```

內建Debug功能：
按下 T 鍵可扣掉三秒，方便測試時間功能。

void addTime(float seconds)

gameTimer是以畫格數來計算時間，
請在void addTime(float seconds)將秒數以每秒60畫格作轉換。

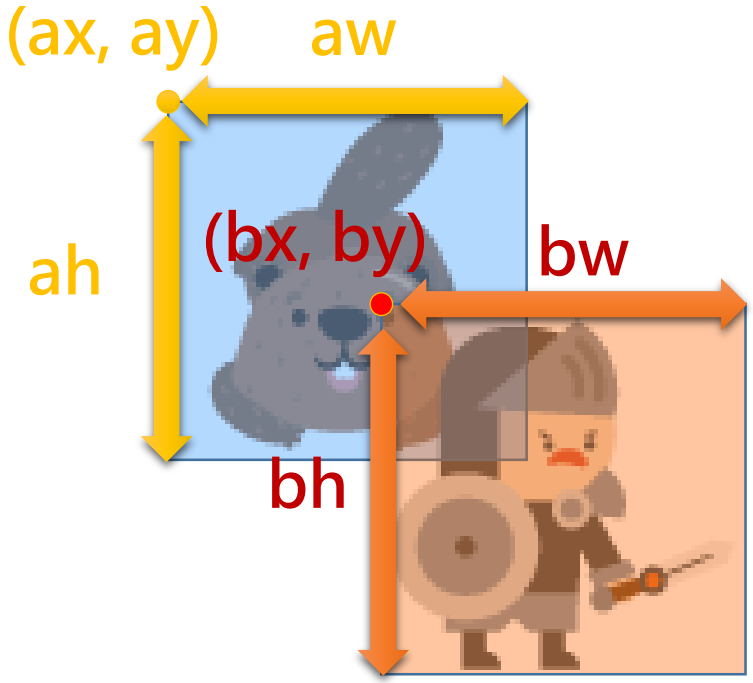
```
final int GAME_INIT_TIMER = 7200;
int gameTimer = GAME_INIT_TIMER;

final float CLOCK_BONUS_SECONDS = 15f;
```

boolean isHit(float ax, float ay, float aw, float ah, float bx, float by, float bw, float bh)

如果四個條件

- A物的左側在B物右側的左方
- A物的右側在B物左側的右方
- A物的上側在B物下側的上方
- A物的下側在B物上側的下方

同時符合，即代表兩物發生碰撞



String convertFramesToTimeString(int frames)

提示：

- 使用nf()函式整理不含十位數的數字，使其符合mm:ss格式
- 小數位無條件捨去

例：3906 frames = 65.1s

1:5 

01:05 

color getTimeTextColor(int frames)

輸入時間	傳回顏色
>= 02:00	#00ffff
01:59 – 01:00	#ffffff
00:59 – 00:30	#ffcc00
00:29 – 00:10	#ff6600
00:09 – 00:00	#ff0000

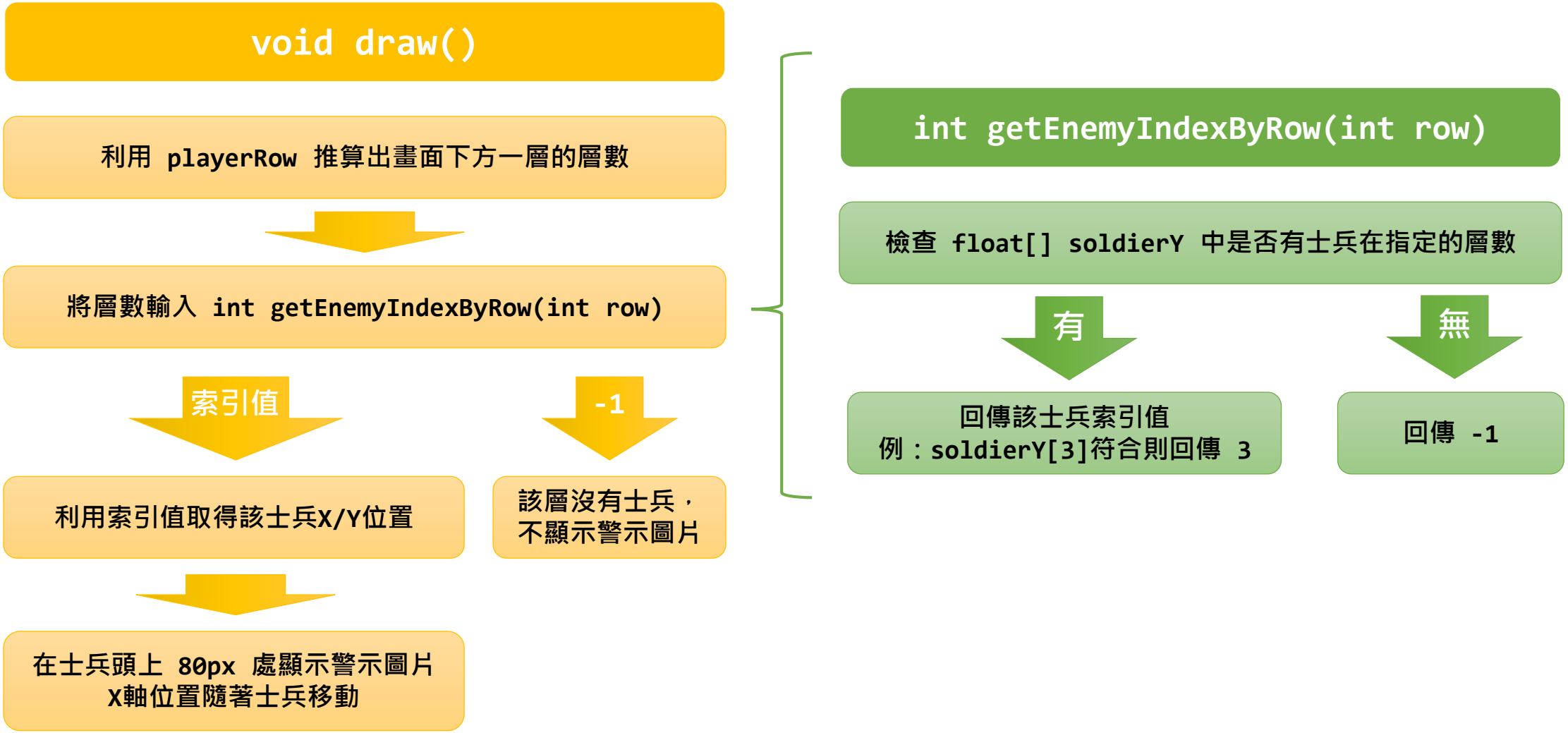
int getEnemyIndexByRow(int row)

(畫面下方一層有士兵時顯示
Caution Sign)

PImage caution

圖片寬高 80 x 80

int getEnemyIndexByRow(int row)



int getEnemyIndexByRow(int row)

由於playerRow是在結束移動時才計算，在下降過程中警示圖片仍會顯示在開始移動前鎖定的敵人頭上，直到土撥鼠抵達下一層時才會消失。

同學只需要利用playerRow來推算畫面下方一層的位置即可完成作業要求效果，不須考慮往下移動過程的顯示問題。

作業要求相關方法

```
182
183 void initCabbages(){
184     cabbageX = new float[6];
185     cabbageY = new float[6];
186
187     for(int i = 0; i < cabbageX.length; i++){
188         cabbageX[i] = SOIL_SIZE * floor(random(SOIL_COL_COUNT));
189         cabbageY[i] = SOIL_SIZE * ( i * 4 + floor(random(4)));
190     }
191 }
192
193 void initClocks(){
194     // Requirement #1: Complete this method based on initCabbages()
195     // - Remember to reroll if the randomized position has a cabbage on the same soil!
196 }
197
```

```
542
543 void addTime(float seconds){                    // Requirement #2
544 }
545
546 boolean isHit(float ax, float ay, float aw, float ah, float bx, float by, float bw, float bh){
547     return false;                               // Requirement #3
548 }
549
550 String convertFramesToTimeString(int frames){  // Requirement #4
551     return "";
552 }
553
554 color getTimeTextColor(int frames){             // Requirement #5
555     return #ffffff;
556 }
557
558 int getEnemyIndexByRow(int row){                // Requirement #6
559
560         // HINT:
561         // - If there's a soldier in that row, return that soldier's index in soldierX/soldierY
562         // (for example, if soldierY[3] is in that row, return 3)
563         // - Return -1 if there's no soldier in that row
564
565     return -1;
566 }
567
568 void drawCaution(){                             // Requirement #6
569
570     // Draw a caution sign above the enemy under the screen using int getEnemyIndexByRow(int row)
571
572         // HINT:
573         // - Use playerRow to calculate the row below the screen
574         // - Use the returned value from int getEnemyIndexByRow(int row) to get the soldier's position from soldierX/soldierY arrays
575         // - Don't draw anything if int getEnemyIndexByRow(int row) returns -1
576 }
577
```

本次作業需要宣告的方法都已經宣告在預設程式碼中，
可以直接修改並參考註解說明。

注意：
預設程式碼並沒有實際呼叫這些方法，
請同學參考註解在正確的位置呼叫使用

宣告位置：
initClocks: 第 193 行
addTime: 第 543 行
isHit: 第 546 行
convertFramesToTimeString: 第 550 行
getTimeTextColor: 第 554 行
getEnemyIndexByRow: 第 558 行
drawCaution: 第 568 行

Demo 影片

https://www.youtube.com/watch?v=YKfIZZ6qHg0