**組別 : 第13組**

**題目 : 很飢餓的遊戲!!!**

**組員 : 106820003 姓名：潘建蒼**

**106820046 姓名：凃昱安**

**指導老師 : 陳偉凱老師**









內容

[一 簡介 4](#_Toc43551334)

[1. 動機 4](#_Toc43551335)

[2. 分工 4](#_Toc43551336)

[二 遊戲介紹 5](#_Toc43551337)

[1. 遊戲說明 5](#_Toc43551338)

[I. RPG遊戲的定義 5](#_Toc43551339)

[II. 什麼是很飢餓的遊戲！！！ 5](#_Toc43551340)

[III. The setting in Don't Starve 5](#_Toc43551341)

[2. 遊戲圖形 8](#_Toc43551342)

[3. 遊戲音效 12](#_Toc43551343)

[三 程式設計 13](#_Toc43551344)

[1. 程式架構 13](#_Toc43551345)

[2. 程式類別 13](#_Toc43551346)

[4. 3.程式技術 18](#_Toc43551347)

[四 結語 18](#_Toc43551348)

[IV. 問題及解決方法 18](#_Toc43551349)

[V. 時間表(不含上課時間) 18](#_Toc43551350)

[VI. 貢獻比例 19](#_Toc43551351)

[VII. 自我檢核表 19](#_Toc43551352)

[VIII. 收獲 20](#_Toc43551353)

[IX. 心得 20](#_Toc43551354)

[X. 對於本課程的建議 20](#_Toc43551355)

[附錄 20](#_Toc43551356)

[I. arror\_attack.js 20](#_Toc43551357)

[II. backpack.js 21](#_Toc43551358)

[III. bat\_wing.js 32](#_Toc43551359)

[IV. bombMan.js 32](#_Toc43551360)

[V. building\_remance.js 44](#_Toc43551361)

[VI. character\_description.js 45](#_Toc43551362)

[VII. clock.js 48](#_Toc43551363)

[VIII. constants.js 51](#_Toc43551364)

[IX. creation\_blood\_status.js 51](#_Toc43551365)

[X. define.js 54](#_Toc43551366)

[XI. drama0.js 55](#_Toc43551367)

[XII. equipmentBar.js 57](#_Toc43551368)

[XIII. fishing.js 58](#_Toc43551369)

[XIV. flying\_arror.js 60](#_Toc43551370)

[XV. food\_base.js 63](#_Toc43551371)

[XVI. game\_object\_detail.js 63](#_Toc43551372)

[XVII. gameOver.js 66](#_Toc43551373)

[XVIII. handle\_game\_over.js 67](#_Toc43551374)

[XIX. handle\_initial\_character.js 69](#_Toc43551375)

[XX. item\_armor.js 76](#_Toc43551376)

[XXI. item\_bee\_sting.js 77](#_Toc43551377)

[XXII. item\_berry.js 77](#_Toc43551378)

[XXIII. item\_black\_berry.js 78](#_Toc43551379)

[XXIV. item\_blank.js 78](#_Toc43551380)

[XXV. item\_branch.js 79](#_Toc43551381)

[XXVI. item\_bush.js 79](#_Toc43551382)

[XXVII. item\_bush\_dig.js 80](#_Toc43551383)

[XXVIII. item\_bush\_growed\_dig.js 81](#_Toc43551384)

[XXIX. item\_camp.js 82](#_Toc43551385)

[XXX. item\_campfire.js 83](#_Toc43551386)

[XXXI. item\_droplet.js 84](#_Toc43551387)

[XXXII. item\_eyeball.js 84](#_Toc43551388)

[XXXIII. item\_fake.js 85](#_Toc43551389)

[XXXIV. item\_firebundle.js 85](#_Toc43551390)

[XXXV. item\_fish.js 87](#_Toc43551391)

[XXXVI. item\_flint.js 87](#_Toc43551392)

[XXXVII. item\_flower.js 88](#_Toc43551393)

[XXXVIII. item\_flower\_dig.js 89](#_Toc43551394)

[XXXIX. item\_flower\_growed\_dig.js 90](#_Toc43551395)

[XL. item\_flower\_picked.js 90](#_Toc43551396)

[XLI. item\_gold.js 91](#_Toc43551397)

[XLII. item\_grass.js 92](#_Toc43551398)

[XLIII. item\_grass\_dig.js 93](#_Toc43551399)

[XLIV. item\_grass\_growed\_dig.js 93](#_Toc43551400)

[XLV. item\_grass\_picked.js 94](#_Toc43551401)

[XLVI. item\_helmat.js 95](#_Toc43551402)

[XLVII. item\_honey.js 96](#_Toc43551403)

[XLVIII. item\_ice.js 96](#_Toc43551404)

[XLIX. item\_king\_wand.js 97](#_Toc43551405)

[L. item\_lamp.js 98](#_Toc43551406)

[LI. item\_meat.js 99](#_Toc43551407)

[LII. item\_monster\_meat.js 99](#_Toc43551408)

[LIII. item\_pigskin.js 100](#_Toc43551409)

[LIV. item\_rope.js 101](#_Toc43551410)

[LV. item\_sapling.js 102](#_Toc43551411)

[LVI. item\_sapling\_dig.js 103](#_Toc43551412)

[LVII. item\_sapling\_growed\_dig.js 103](#_Toc43551413)

[LVIII. item\_snow\_ball.js 104](#_Toc43551414)

[LIX. item\_space\_wand.js 105](#_Toc43551415)

[LX. item\_spider\_web.js 106](#_Toc43551416)

[LXI. item\_stone.js 107](#_Toc43551417)

[LXII. item\_tree\_dig.js 108](#_Toc43551418)

[LXIII. item\_waikei\_homework.js 109](#_Toc43551419)

[LXIV. item\_wood.js 109](#_Toc43551420)

[LXV. item\_wood\_armor.js 110](#_Toc43551421)

[LXVI. level\_up\_animation.js 111](#_Toc43551422)

[LXVII. level2\_change.js 112](#_Toc43551423)

[LXVIII. loadGame.js 113](#_Toc43551424)

[LXIX. Local\_map\_0.js 117](#_Toc43551425)

[LXX. magic\_attack.js 134](#_Toc43551426)

[LXXI. mainGame.js 134](#_Toc43551427)

[LXXII. map\_item\_tree.js 135](#_Toc43551428)

[LXXIII. map\_selector.js 137](#_Toc43551429)

[LXXIV. monster\_base.js 139](#_Toc43551430)

[LXXV. monster\_bat.js 146](#_Toc43551431)

[LXXVI. monster\_bee.js 147](#_Toc43551432)

[LXXVII. monster\_boss.js 147](#_Toc43551433)

[LXXVIII. monster\_cow.js 161](#_Toc43551434)

[LXXIX. monster\_cute\_little\_eye.js 161](#_Toc43551435)

[LXXX. monster\_pig.js 162](#_Toc43551436)

[LXXXI. myMenu.js 163](#_Toc43551437)

[LXXXII. myMenu\_gameteam.js 166](#_Toc43551438)

[LXXXIII. myMenu\_tutorial.js 168](#_Toc43551439)

[LXXXIV. normal\_attack.js 170](#_Toc43551440)

[LXXXV. npc\_event.js 171](#_Toc43551441)

[LXXXVI. npc1.js 177](#_Toc43551442)

[LXXXVII. npc2.js 181](#_Toc43551443)

[LXXXVIII. null\_attack.js 184](#_Toc43551444)

[LXXXIX. object\_factory.js 185](#_Toc43551445)

[XC. player\_hide\_animation.js 186](#_Toc43551446)

[XCI. proxy.js 186](#_Toc43551447)

[XCII. reduceDurabilityVisitor.js 201](#_Toc43551448)

[XCIII. score.js 202](#_Toc43551449)

[XCIV. skill\_handler.js 204](#_Toc43551450)

[XCV. skill\_timer.js 205](#_Toc43551451)

[XCVI. spear\_handler.js 206](#_Toc43551452)

[XCVII. synthesisBar.js 207](#_Toc43551453)

[XCVIII. tool.js 220](#_Toc43551454)

[XCIX. weapon.js 220](#_Toc43551455)

[C. world\_map.js 221](#_Toc43551456)

# 一 簡介

1. 動機

物件導向實習課程以遊戲製作作為練習題目，實作一個自行選擇的遊戲程式，藉由開發遊戲程式練習物件間的互動，演練繼承與多行等物件關係，並套用設計樣式提升程式品質。本專案選擇Don’t starve作為模仿對象，因其具備遊玩自由度高且物件繁多的特性，適合作為練習物件導向程式設計的對象。

1. 分工

每週於上課時討論上週進度是否有bug或行為需要修正，若有則加入本週進度，並由上週負責者繼續完成，加完修正類的工作後，會考量修正類工作的份量，新增一些新功能的實作，通常會由提案者獨自完成，若遇到與其他物dependency較高的物件或較複雜的物件，會視情況採取pair programming。

# 二 遊戲介紹

1. 遊戲說明
2. RPG遊戲的定義

RPG(角色扮演)遊戲是一種遊戲類型，在遊戲中，玩家扮演虛擬世界中的一個或者幾個角色進行遊戲，玩家通過操控遊戲角色與敵人戰鬥，提升等級、收集裝備和完成遊戲設置的任務，並體驗劇情。通常這類遊戲都是由玩家扮演角色在遊戲世界中漫遊，而一路上的各種遭遇則是玩家人物成長及遊戲進行的重要關鍵所在。

1. 什麼是很飢餓的遊戲！！！

以生存為主軸的2D 冒險養成RPG遊戲，在遊戲中玩家必須想辦法活下去，並在這過程中提升自己的裝備和人物數值，最後目標是打倒最終魔王，才算通關。

1. The setting in Don't Starve

* **功能操作：**

|  |  |
| --- | --- |
| 人物動作 | 操作說明 |
| 撿取 | 對著物品按空白鍵 |
| 丟棄 | 滑鼠需移至背包欲丟棄物品上，按D鍵 |
| 對話 | 對著npc按空白鍵 |
| 普攻(無裝備武器) | 按S鍵 |
| 突刺(裝備長矛) | 按S鍵 |
| 射箭(裝備弓箭) | 按S鍵 |
| 傳送(裝備空間法杖) | 按S鍵 |
| 放魔法(裝備火法杖  、冰法杖) | 長按S鍵，直到詠唱完畢 |
| 吃食物 | 點擊背包中的食物 |
| 裝裝備 | 點擊背包中的裝備 |
| 合成 | 點擊合成欄位的合成紐 |
| 釣魚 | 裝備釣竿，對著池塘按F鍵，當人物出現'!'時，按空白鍵 |
| 挖植物 | 裝備鏟子，對著植物按空白鍵，適用目標：花、草、樹、樹苗和漿果 |
| 種植物 | 點擊背包中的植物(須為挖掘過的植物) |
| 砍樹 | 裝備斧頭，對著樹木按空白鍵 |
| 挖礦 | 裝備十字搞，對著岩石按空白鍵 |
| 查看角色狀態 | 按E鍵 |

* **金手指：**

|  |  |
| --- | --- |
| 密技 | 操作方式 |
| 無敵 | 按P鍵 |
| 自殺 | 按Q鍵 |
| 直接勝利 | 按W鍵 |
| 召喚蜜蜂 | 按R鍵 |
| 召喚豬豬 | 按T鍵 |
| 召喚牛牛 | 按Y鍵 |
| 召喚眼球 | 按U鍵 |
| 召喚蝙蝠 | 按I鍵 |
| 召喚BOSS | 按O鍵 |
| 大祕寶 | 往左走 |

* **角色能力設置：**

|  |  |
| --- | --- |
| 能力 | 說明 |
| 生命 | 當前血量 |
| 力量 | 血量上限 |
| 魔力 | 當前魔力 |
| 智力 | 魔力上限 |
| 物功 | 物理攻擊數值 |
| 魔功 | 魔法攻擊數值 |
| 弓功 | 弓攻擊數值 |
| 防禦 | 防禦素質 |

* **怪物設置：**

|  |  |  |
| --- | --- | --- |
| 怪獸名稱 | 招式 | 掉落物 |
| 眼球怪 | 衝撞 | 眼球、露水 |
| 豬人 | 衝撞 | 豬皮、怪物肉 |
| 牛人 | 衝撞 | 怪物肉、蜘蛛網 |
| 蝙蝠怪 | 衝撞 | 蝙蝠翅膀、怪物肉 |
| 蜜蜂怪 | 衝撞 | 毒刺、蜂蜜 |
| 魔王 | 順移、死亡空間、丟飛碟、魔力爪、藤蔓束縛、反擊 |  |

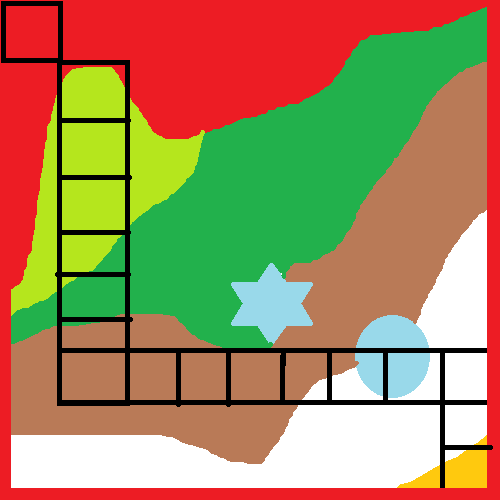
* **武器設置：**

|  |  |
| --- | --- |
| 武器名稱 | 效果說明 |
| 長矛 | 普功+特效 |
| 弓箭 | 射出一支箭，範圍是5格 |
| 火法杖 | 施放前須詠唱；效果為對著角色前方3\*3格子內所有敵人產生爆炸，傷害為40點傷害 |
| 冰法杖 | 施放前須詠唱；效果為對著角色前方3\*3格子內所有敵人產生結凍，傷害為40點傷害 |
| 空間法杖 | 施放前須詠唱；效果為瞬間移動至魔王所在位置 |
| 國王法杖 | 施放前須詠唱；效果為把角色偽裝成灌木叢，避免怪物發現他，但對魔王沒用，  持續時間為10秒 |

* **遊戲主線：**

|  |  |
| --- | --- |
|  | 說明 |
| 流程一 | 觀看開場劇情 |
| 流程二 | 找小丑哥哥觸發漿果任務 |
| 流程三 | 完成後去找商人莉莉觸發後續任務 |
| 流程四 | 蒐集物資、鍛鍊等級、合成武器 |
| 流程五 | 打倒魔王 |
|  | 遊戲結束 |

* **遊戲地圖概觀：**

****

1. 遊戲圖形

|  |  |
| --- | --- |
| 遊戲主畫面 | 遊戲操作說明頁面 |
| 未命名.png | 未命名.png |
| 遊戲團隊頁面 | **遊戲前置劇情** |
| 未命名.png | 未命名.png |
| 遊戲創角頁面 | **豬人與蜜蜂怪** |
| 未命名.png |  |
| 眼球怪與牛人 | **蝙蝠怪與魔王** |
|  |  |

|  |  |
| --- | --- |
| 小丑哥哥對話畫面 | 遺跡外部畫面 |
|  |  |
| 遺跡內部畫面 | **莉莉對話畫面** |
|  |  |
| 採摘植物 | **發射弓箭** |
|  |  |
| 國王法杖隱身特效 | **空間法杖傳送特效** |
|  |  |

|  |  |
| --- | --- |
| 合成欄 | 魔王近戰特效及束縛技能 |
|  |  |
| 魔王範圍攻擊特效 | **指向背包中物品顯示說明** |
|  |  |
| 開鑿岩石 | **砍樹** |
|  |  |
| 升級畫面 | **釣魚及魚上鉤提示** |
|  |  |

|  |  |
| --- | --- |
| 挖起植物 | 長矛特效 |
|  |  |
| 冰、火法杖詠唱條 | **火法杖特效** |
|  |  |
| 冰法杖特效 | **遊戲死亡畫面** |
|  |  |
| 遊戲通關畫面 | **分數結算** |
|  |  |

1. 遊戲音效

|  |  |
| --- | --- |
| 名稱 | 使用場景 |
| Hot\_Heat.mp3 | 早上播放 |
| 遊戲王經典配樂.mp3 | 下午播放 |
| night.mp3 | 晚上播放 |
| levelup.mp3 | 升級時播放 |
| 刀劍神域op1《crossing field》鋼琴曲.mp3 | 遊戲教學頁面和小組頁面播放 |
| 女慘叫.mp3 | 主角死亡時播放 |
| 火魔法.mp3 | 施放火法術時播放 |
| 冰魔法.mp3 | 施放冰法術時播放 |
| 長矛.mp3 | 長矛突刺時播放 |
| 弓箭.mp3 | 弓箭射擊時播放 |
| 牛叫聲.mp3 | 牛人走路時播放 |
| 打擊.mp3 | 角色普通攻擊時播放 |
| 怪物攻擊.mp3 | 怪獸攻擊時播放 |
| 按按鍵.mp3 | 點擊時播放 |
| 挖掘.mp3 | 挖掘時播放 |
| 砍樹.mp3 | 砍樹時播放 |
| 眼球叫聲.mp3 | 眼球怪走路時播放 |
| 釣魚.mp3 | 釣魚時播放 |
| 勝利.mp3 | 分數結算時播放 |
| 開門.mp3 | 通關或死亡後，接續到分數結算前播放 |
| 開場.mp3 | 開場劇情播放 |
| 順移.mp3 | 魔王和角色瞬間移動時播放 |
| 綑綁.mp3 | 魔王施展綑綁技能時播放 |
| 遊戲開場音樂.mp3 | 進入遊戲主選單頁面時播放 |
| 摔東西.mp3 | 魔王施展丟飛碟技能時播放 |
| 範圍攻擊.mp3 | 魔王施展死亡空間時播放 |
| 蝙蝠叫聲.mp3 | 蝙蝠走路時播放 |
| 豬叫聲.mp3 | 豬走路時播放 |
| 撿取.mp3 | 撿取時播放 |
| 魔王叫聲.mp3 | 魔王走路時播放 |

# 三 程式設計

1. 程式架構
2. 程式類別

|  |  |  |
| --- | --- | --- |
| 1. 類別名稱 | 檔行數(js) | 說明 |
| Local\_map\_0.js | 4355 | 包含所有地形的資料及處理 |
| arror\_attack.js | 43 | 弓箭的功能 |
| backpack.js | 400 | 背包的功能，用於存放獲得之物件 |
| bat\_wing.js | 10 | 食物，掉落自蝙蝠怪 |
| bombMan.js | 455 | 玩家操控的角色 |
| building\_remance.js | 28 | 遺跡物件、功能 |
| character\_description.js | 142 | 角色面板，用於顯示角色當前各項數值 |
| clock.js | 95 | 顯示時間及天數，一天分成白天、下午、夜晚 |
| constants.js | 14 | 包含開發所需之常數 |
| creation\_blood\_status.js | 88 | 顯示角色當前血量、魔力、飢餓度，顯示畫面中怪物的血量 |
| define.js | 58 | 定義圖片、音訊等資料的路徑 |
| drama0.js | 55 | 開場劇情 |
| equipmentBar.js | 51 | 角色裝備欄，分為頭部、身體、手 |
| fishing.js | 48 | 處理釣魚，包括等待時間、上鉤時間 |
| flying\_arror.js | 111 | 弓箭，處理射速及傷害 |
| food\_base.js | 25 | 所有食物物件的Base class |
| gameOver.js | 35 | 遊戲的其中一個階段，處理遊戲結束的階段 |
| game\_object\_detail.js | 113 | 用於顯示背包中物件的資料 |
| handle\_game\_over.js | 67 | 處理遊戲結束時的畫面，及分數結算畫面 |
| handle\_initial\_character.js | 242 | 處理遊戲開始時的角色初始化 |
| item\_armor.js | 34 | 防具 |
| item\_bee\_sting.js | 28 | 蜜蜂怪掉落物，用於合成 |
| item\_berry.js | 10 | 食物，由漿果叢生成 |
| item\_black\_berry.js | 10 | 食物，壞掉的漿果 |
| item\_blank.js | 15 | Null object，所有物件在地圖上的位置會對應陣列中位置，因此即使是空的也需要創立blank物件 |
| item\_branch.js | 25 | 樹枝，用於合成，由小樹苗產生 |
| item\_bush.js | 39 | 漿果叢，可採摘漿果 |
| item\_bush\_dig.js | 25 | 採摘完後用鏟子挖起漿果叢，可獲得此物件，用於移植漿果叢 |
| item\_bush\_growed\_dig.js | 25 | 採摘前用鏟子挖起漿果叢，可獲得此物件，用於移植漿果叢 |
| item\_camp.js | 28 | 帳篷，無功能，裝飾用 |
| item\_campfire.js | 28 | 營火，無功能，裝飾用 |
| item\_droplet.js | 28 | 水滴，無功能，裝飾用，掉落自眼球怪 |
| item\_eyeball.js | 10 | 食物，掉落自眼球怪 |
| item\_fake.js | 13 | Null object，用於處理體積大於一格的物件，不具圖案但有碰撞 |
| item\_firebundle.js | 45 | 火把，無功能，裝飾用 |
| item\_fish.js | 10 | 食物，可由釣魚獲得 |
| item\_flint.js | 28 | 燧石，合成用物件，可在地上撿取或開鑿石頭獲得 |
| item\_flower.js | 36 | 小花，用於採摘花朵 |
| item\_flower\_dig.js | 25 | 採摘完後用鏟子挖起小花，可獲得此物件，用於移植小花 |
| item\_flower\_growed\_dig.js | 27 | 採摘前用鏟子挖起小花，可獲得此物件，用於移植小花 |
| item\_flower\_picked.js | 10 | 食物，花朵，可由採摘小花獲得 |
| item\_gold.js | 28 | 金塊，合成用物件，開鑿石頭獲得 |
| item\_grass.js | 40 | 草，可用於採摘乾草 |
| item\_grass\_dig.js | 27 | 採摘完後用鏟子挖起草，可獲得此物件，用於移植草 |
| item\_grass\_growed\_dig.js | 27 | 採摘前用鏟子挖起草，可獲得此物件，用於移植草 |
| item\_grass\_picked.js | 25 | 乾草，合成用物件，可由採摘草獲得 |
| item\_helmat.js | 33 | 防具 |
| item\_honey.js | 10 | 食物，掉落自蜜蜂怪 |
| item\_ice.js | 28 | 合成用物件，由雪球合成 |
| item\_king\_wand.js | 35 | 國王法杖，使用後變身漿果叢一段時間，期間怪物不會追逐玩家 |
| item\_lamp.js | 29 | 提燈，無功能，裝飾用 |
| item\_meat.js | 10 | 食物，大肉，掉落自豬怪、牛怪 |
| item\_monster\_meat.js | 10 | 食物，怪物肉，掉落自蝙蝠怪 |
| item\_pigskin.js | 28 | 食物，豬皮，掉落自豬怪 |
| item\_rope.js | 28 | 合成用物件，可由草合成 |
| item\_sapling.js | 36 | 小樹苗，用於採摘小樹枝 |
| item\_sapling\_dig.js | 25 | 採摘完後用鏟子挖起小樹苗，可獲得此物件，用於移植小樹苗 |
| item\_sapling\_growed\_dig.js | 25 | 採摘前用鏟子挖起小樹苗，可獲得此物件，用於移植小樹苗 |
| item\_snow\_ball.js | 28 | 合成用物件，隨機生成於雪地地形 |
| item\_space\_wand.js | 35 | 空間法杖，可直接傳送到岩漿區 |
| item\_spider\_web.js | 29 | 合成用物件，掉落自牛怪 |
| item\_stone.js | 35 | 多出現於岩石地形，可用十字鎬開鑿 |
| item\_tree\_dig.js | 25 | 砍倒樹後，可用鏟子挖起獲得，用於移植樹 |
| item\_waikei\_homework.js | 28 | 合成用物件 |
| item\_wood.js | 28 | 合成用物件，可由砍伐樹木獲得 |
| item\_wood\_armor.js | 33 | 盔甲，防禦用 |
| level2\_change.js | 50 | 控制關卡的流程，init、draw、load |
| level\_up\_animation.js | 15 | 升級動畫 |
| loadGame.js | 153 | 載入遊戲檔案 |
| magic\_attack.js | 21 | 攻擊時會傳入攻擊的方法讓玩家執行，magic\_ attack為魔法攻擊成功時傳進player的物件 |
| mainGame.js | 13 | 關卡流程控制 |
| map\_item\_tree.js | 79 | 樹的物件，砍伐可獲得木頭 |
| map\_selector.js | 66 | 處理大地圖的運算和判斷 |
| monster\_base.js | 273 | 怪物的Base class，除了boss外，皆繼承此物件 |
| monster\_bat.js | 23 | 蝙蝠怪，繼承Monster  \_base |
| monster\_bee.js | 23 | 蜜蜂怪，繼承Monster  \_base |
| monster\_boss.js | 494 | 魔王，打倒魔王後即通關，具一進戰攻擊、兩遠程攻擊、一反彈攻擊、一束縛技能 |
| monster\_cow.js | 23 | 牛怪，繼承Monster\_  base |
| monster\_cute\_little\_eye.js | 23 | 眼球怪，繼承Monster  \_base |
| monster\_pig.js | 23 | 豬怪，繼承Monster\_  base |
| myMenu.js | 129 | 遊戲主選單 |
| myMenu\_gameteam.js | 61 | 介紹遊戲團隊 |
| myMenu\_tutorial.js | 83 | 介紹遊戲玩法和指令 |
| normal\_attack.js | 21 | 攻擊時會傳入攻擊的方法，讓玩家執行，normal \_attack為近戰攻擊成功時傳進player的物件 |
| npc1.js | 132 | 小丑哥哥的NPC，負責NPC的移動和邏輯判斷 |
| npc2.js | 128 | 商人莉莉的NPC，負責NPC的移動和邏輯判斷 |
| npc\_event.js | 238 | 使發布任務的角色頭上發出小提示；儲存已寫好劇本，並引導玩家 |
| null\_attack.js | 6 | 攻擊時會傳入攻擊的方法，讓玩家執行，null\_ attack為攻擊失敗時傳進player的物件 |
| object\_factory.js | 34 | 用於構建物件的Factory |
| player\_hide\_animation.js | 16 | 處理國王法杖使用時的動畫 |
| proxy.js | 8833 | 判斷角色目前所在的小地圖的物件是否創建，若無創建則建立之 |
| reduceDurabilityVisitor.js | 25 | 用於物件減少耐久的visitor |
| score.js | 73 | 紀錄分數 |
| skill\_handler.js | 46 | 處理法杖施放動畫 |
| skill\_timer.js | 39 | 處理法杖集氣所需時間及動畫 |
| spear\_handler.js | 20 | 處理使用長矛的特效 |
| synthesisBar.js | 480 | 合成欄，用於合成用生存用物件 |
| tool.js | 31 | 所有工具物件的Base class |
| weapon.js | 32 | 所有攻擊類武器物件的Base class |
| world\_map.js | 1229 | 遊戲主地圖，含有遊戲大部分物件 |
| 總行數 | 20415 |  |

3.程式技術

# 四 結語

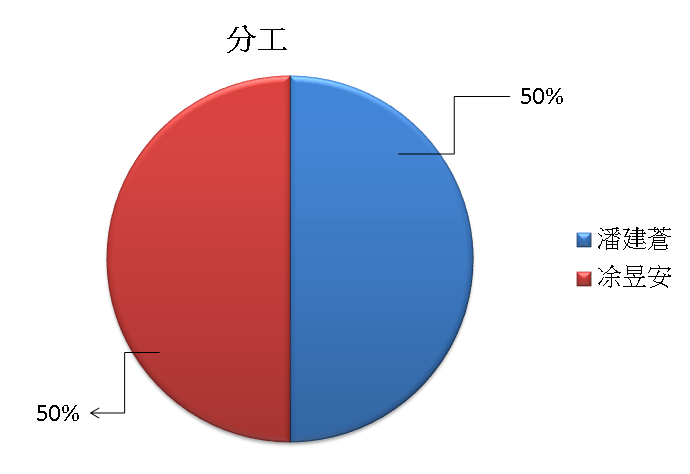
1. 問題及解決方法

|  |  |  |
| --- | --- | --- |
|  | 問題 | 解決方法 |
| 1 | 當初創建遊戲大地圖時採用400\*400的二維陣列來存取地圖資料，然而在遊戲載入後，發現會產生嚴重的delay |  |
| 2 | 當遊戲選單中的開始遊戲按鍵一被按下，遊戲地圖即開始建立，建置完成後遊戲才開始；然而地圖物件底下包含100張遊戲小地圖物件，十分龐大，等待建置完成的時間約為5至6分鐘，嚴重危害到使用者體驗 | 使用proxy pattern |
| 3 | 在使用git的時候，常發生conflict的現象，會帶來開發上的不方便和時間上的浪費 |  |
| 4 | JS語言在變數的創建太過自由，常不小心打錯變數的名字，導致又創建了一個新的變數，造成不易除錯的困境 |  |
| 5 | 無法確定物件的draw在框架底下何時會被呼叫，使物件的繪製不符合預期 | 需要掌控draw時機的物件，我們改用自己寫的方法來控制。例如：為了使Clock物件在計時的畫面達成連續感，而使用SetInterval的方式讓它自行呼叫。 |
| 6 | 在Canvas上繪出遊戲物件需要指定(x, y)座標；為了使物件畫在指定的範圍上，通常的做法是先隨便給個座標，再透過觀察法慢慢調整至目標位置。常常浪費需多時間在調整上 |  |
| 7 | 在繪製物件常發現一個現象，當持續呼叫draw的時候，過去所繪製的物件不會被清掉，而在畫面上留下殘餘的部分，造成畫面稜亂，遊戲體驗差 |  |
| 8 | 在同一個物件底下宣告this.audio播放音樂時，會發現audio.play()呼叫的方法和宣告的地方不在同一個方法中時，會觸發錯誤 |  |
| 9 | 怪物的移動一開始是根據格座標，無法平滑的進行移動 |  |
| 10 | 我們所寫的是RPG遊戲，所以會需要很多的遊戲物件；但是很多的物件其底下包含了很多相同的變數和方法，在結構上大致相同，只會有些 許差異。但我們卻要為他們分別創建不同的物件類別，結果便是創建了數十種物件類別，導致程式碼看起來十分混亂 | Factory Pattern |
| 11 | 遊戲中有些相似類別，類別中都有部分方法內容相似，但還是需要判斷目前要做事的類別才能正確呼叫對應的適當類別，例如:減少裝備耐 久度的方法會因裝備的不同而有差異 | Visitor Pattern |
| 12 | 部分遊戲中類別的函式會根據條件進行不同的行為，例如: 角色物件下有攻擊函式，但攻擊的方法會隨持有的武器而改變 | Strategy Pattern |
| 13 | 我們會根據角色使用的武器給定attackMode，並傳送給角色，由角色來進行攻擊，但在某些情況下，攻擊會失敗 或無法發動，例如:魔法類武器沒有集滿能量便發動攻擊，那麼此種情形下該傳入何種attackMode便成為一個問題 | Null Object Pattern |

1. 時間表(不含上課時間)

|  |  |  |  |
| --- | --- | --- | --- |
| 週次 | 凃昱安(小時) | 潘建蒼(小時) | 說明 |
| 1 | 18 | 18 | 設計遊戲地圖並呈現畫面 |
| 2 | 10 | 10 | 人物移動動畫、部分地圖物件 |
| 3 | 7 | 6 | 撿取物件功能、背包系統、遊戲主選單製作、人物狀態(飢餓度、生命)製作 |
| 4 | 10 | 12 | 角色死亡功能、時間系統、合成欄位製作、合成功能製作 |
| 5 | 14 | 12 | 新增裝備欄、部分食物、 遊戲合成用物件、死亡後回到menu |
| 6 | 7 | 10 | 斧頭功能(砍樹)、十字搞功能(敲岩石)、人物屬性面板、角色升級系統、角色升級音效 |
| 7 | 11 | 12 | 鏟子功能(挖植物)、火法杖功能、丟棄包包內容物、小怪、種植植物 |
| 8 | 11 | 12 | 隨機產生怪、魔法傷害、近戰武器傷害、弓箭傷害、攻擊特效、角色死亡動畫、材料及物品說明 |
| 9 | 16 | 10 | 小樹枝植物(採、挖、種)、怪物朝人移動、怪物攻擊人、角色升級動畫 |
| 10 | 4 | 2 | 怪物轉向、修角色面板能力bug、人物攻擊修正 |
| 11 | 4 | 10 | 完整地圖、釣魚系統、NPC系統、受傷動畫 |
| 12 | 7 | 6 | 修改完整地圖、NPC系統 |
| 13 | 6 | 4 | 法杖製作、創角功能 |
| 14 | 17 | 14 | 加入防具、計天數、地圖編輯、初始劇情、任務1、新增商人莉莉遺跡物件製作、魔王、怪物(牛、豬、蜂、蝙蝠)、怪物硬直、怪物受傷動畫、怪物掉落物、食物(眼球、蝙蝠翅、蜂蜜、大肉、怪物肉)、分數畫面、通關畫面 |
| 15 | 15 | 15 | 魔王攻擊傷害、物件地圖編輯、新手任務編輯、打報告、發布任務、音效、隨機生成物件、魔王死亡通關事件觸發 |
| 合計 | 157 | 155 |  |

1. 貢獻比例



1. 自我檢核表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 項目 | 完成否 | 無法完成的原因 |
| 1 | 解決 Memory leak | □已完成 ☑未完成 | JS專案無此項目 |
| 2 | 自定遊戲 Icon | □已完成 ☑未完成 | JS專案無此項目 |
| 3 | 全螢幕啟動 | ☑已完成 □未完成 |  |
| 4 | 有 About 畫面 | ☑已完成 □未完成 |  |
| 5 | 初始畫面說明按鍵及滑鼠 之用法與密技 | ☑已完成 □未完成 |  |
| 6 | 上傳 setup/apk/source 檔 | □已完成 ☑未完成 | JS專案無此項目 |
| 7 | setup 檔可正確執行 | □已完成 ☑未完成 | JS專案無此項目 |
| 8 | 報告字型、點數、對齊、行 距、頁碼等格式正確 | ☑已完成 □未完成 |  |
| 9 | 報告封面、側邊格式正確 | ☑已完成 □未完成 |  |
| 10 | 報告附錄程式格式正確 | ☑已完成 □未完成 |  |

1. 收獲
2. 心得
3. 對於本課程的建議

# 附錄

1. arror\_attack.js

var Arror\_attack = function(player, monster) {

    this.player = player;

    this.monster = monster;

    this.attackSuccess = false;

    this.arrorHandle = [];

    this.playerWalkDirection;

    this.playerPositionOnMap;

    this.visitor = new ReduceDurabilityVisitor();

    this.setPositionAndDirection = function(playerWalkDirection, playerPositionOnMap){

        this.playerWalkDirection = playerWalkDirection;

        this.playerPositionOnMap = playerPositionOnMap;

    }

    this.attack = function(){

        this.attackSuccess = false;

        var arror = new Flying\_arror(this.playerWalkDirection, this.playerPositionOnMap, this.monster, this.player.character\_descruption\_total\_point[4]);

        arror.init();

        this.arrorHandle.push(arror);

        this.player.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

    }

    this.update = function(){

        if(this.arrorHandle.length != 0){

            for(var i = 0;i < this.arrorHandle.length;i++){

                this.arrorHandle[i].update();

            }

            var i = 0;

            while(i < this.arrorHandle.length) {

                if(this.arrorHandle[i].attackEnd){

                    this.arrorHandle[i].draw(Framework.Game.\_context);

                    this.arrorHandle.splice(i, 1);

                }else{

                    i++;

                }

            }

        }

    }

    this.draw = function(ctx){

        if(this.arrorHandle.length != 0){

            for(var i = 0;i < this.arrorHandle.length;i++){

                this.arrorHandle[i].draw(ctx);

            }

        }

    }

};

1. backpack.js

var Backpack = function() {

    this.backpack = new Framework.Sprite(define.materialPath + 'backpack.png');

    this.backpack.scale = 2;

    this.objectFactory = new Object\_factory();

    this.backpackPosition = [];

    this.objectPosition = [];

    var berry = new Item\_berry();

    berry.amount = 100;

    var branch = new Item\_branch();

    branch.amount = 100;

    var flint = new Item\_flint();

    flint.amount = 100;

    var gold = new Item\_gold();

    gold.amount = 100;

    var grass = new Item\_grass\_picked();

    grass.amount = 100;

    var pix = this.objectFactory.createObject(

        {

            type: "Tool",

            imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_pixilart.png'),

            attackPoint: 2,

            itemNumber: 21,

            reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

        }

    );

    this.itemList = [new Item\_wood\_armor(),new Item\_helmat(),berry,branch,flint,gold,grass,pix,new Item\_space\_wand()];

    this.stackableList = [1,2,4,5,6,7,8,9,10,11,12,13,14,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54];

    this.selectedIndex = -1;

    for(var j = 0; j < 17; j++){

        this.backpackPosition.push({x: (j+5)\*64, y: 13\*64});

        this.objectPosition.push({x: (j+5), y: 13});

    }

    this.getSelectedItem= function(){

        if(this.selectedIndex == -1)

            return null;

        if( this.itemList[this.selectedIndex] != null)

            return this.itemList[this.selectedIndex];

        else

            return null;

    }

    this.dropSelectedItem = function(){

        if(this.selectedIndex != -1)

            this.arrayRemoveByIndex(this.selectedIndex);

    }

    this.addItemBySynthesis = function(item\_num){

        var check = this.checkIfStackableExist(item\_num);

        if(check == -1){

            switch(item\_num){

                case 1:

                    this.itemList.push(new Item\_flower());

                    break;

                case 2:

                    this.itemList.push(new Item\_spider\_web());

                    break;

                case 3:

                    this.itemList.push(new Item\_stone());

                    break;

                case 4:

                    this.itemList.push(new Item\_branch());

                    break;

                case 5:

                    this.itemList.push(new Item\_waikei\_homework());

                    break;

                case 6:

                    this.itemList.push(new Item\_grass());

                    break;

                case 7:

                    this.itemList.push(new Item\_wood());

                    break;

                case 8:

                    this.itemList.push(new Item\_flint());

                    break;

                case 9:

                    this.itemList.push(new Item\_gold());

                    break;

                case 10:

                    this.itemList.push(new Item\_pigskin());

                    break;

                case 11:

                    this.itemList.push(new Item\_bee\_sting());

                    break;

                case 12:

                    this.itemList.push(new Item\_snow\_ball());

                    break;

                case 13:

                    this.itemList.push(new Item\_rope());

                    break;

                case 14:

                    this.itemList.push(new Item\_droplet());

                    break;

                case 15:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_pixilart.png'),

                            attackPoint: 1,

                            itemNumber: 15,

                            reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                        }

                    ));

                    break;

                case 16:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_ax.png'),

                            attackPoint: 1,

                            itemNumber: 16,

                            reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                        }

                    ));

                    break;

                case 17:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_fishing\_rod.png'),

                            attackPoint: 1,

                            itemNumber: 17,

                            reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                        }

                    ));

                    break;

                case 18:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_shovel.png'),

                            attackPoint: 1,

                            itemNumber: 18,

                            reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                        }

                    ));

                    break;

                case 19:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_ax.png'),

                            attackPoint: 2,

                            itemNumber: 19,

                            reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

                        }

                    ));

                    break;

                case 20:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_shovel.png'),

                            attackPoint: 2,

                            itemNumber: 20,

                            reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

                        }

                    ));

                    break;

                case 21:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Tool",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_pixilart.png'),

                            attackPoint: 2,

                            itemNumber: 21,

                            reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

                        }

                    ));

                    break;

                case 22:

                    this.itemList.push(new Item\_helmat());

                    break;

                case 23:

                    this.itemList.push(new Item\_armor());

                    break;

                case 24:

                    this.itemList.push(new Item\_wood\_armor());

                    break;

                case 25:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Weapon",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_spear.png'),

                            attack: [3,0,0],

                            itemNumber: 25,

                            scale: 0.8,

                            reduceDurability: function(visitor){visitor.visitSpear(this)}

                        }

                    ));

                    break;

                case 26:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Weapon",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_arror.png'),

                            attack: [0,0,3],

                            itemNumber: 26,

                            scale: 2,

                            reduceDurability: function(visitor){visitor.visitArror(this)}

                        }

                    ));

                    break;

                case 27:

                    this.itemList.push(new Item\_king\_wand());

                    break;

                case 28:

                    this.itemList.push(new Item\_space\_wand());

                    break;

                case 29:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Weapon",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_fire\_wand.png'),

                            attack: [0,2,0],

                            itemNumber: 29,

                            scale: 0.8,

                            reduceDurability: function(visitor){visitor.visitWand(this)}

                        }

                    ));

                    break;

                case 30:

                    this.itemList.push(this.objectFactory.createObject(

                        {

                            type: "Weapon",

                            imagePath: new Framework.Sprite(define.materialPath + 'item\_ice\_wand.png'),

                            attack: [0,2,0],

                            itemNumber: 30,

                            scale: 0.8,

                            reduceDurability: function(visitor){visitor.visitWand(this)}

                        }

                    ));

                    break;

                case 31:

                    this.itemList.push(new Item\_lamp());

                    break;

                case 32:

                    this.itemList.push(new Item\_firebundle());

                    break;

                case 33:

                    this.itemList.push(new Item\_camp());

                    break;

                case 34:

                    this.itemList.push(new Item\_campfire());

                    break;

                case 35:

                    this.itemList.push(new Item\_ice());

                    break;

                case 36:

                    this.itemList.push(new Item\_bush());

                    break;

                case 37:

                    this.itemList.push(new Item\_flower\_picked());

                    break;

                case 38:

                    this.itemList.push(new Item\_grass\_picked());

                    break;

            }

        }else{

            this.itemList[check].amount += 1;

        }

    }

    this.addItemByObject = function(object){

        var check = this.checkIfStackableExist(object.item\_num);

        if(check != -1)

            this.itemList[check].amount += object.amount;

        else

            this.itemList.push(object);

    }

    this.checkIfPickAvailable = function(item\_num){

        if(this.itemList.length < 17)

            return true;

        else if(item\_num == 1){

            for(var i = 0;i < this.itemList.length;i++){

                if(this.itemList[i].item\_num == 37)

                    return true;

            }

        }

        else if(item\_num == 6){

            for(var i = 0;i < this.itemList.length;i++){

                if(this.itemList[i].item\_num == 38)

                    return true;

            }

        }

        else if(item\_num == 36){

            for(var i = 0;i < this.itemList.length;i++){

                if(this.itemList[i].item\_num == 39)

                    return true;

            }

        }

        else if(item\_num == -4){

            for(var i = 0;i < this.itemList.length;i++){

                if(this.itemList[i].item\_num == 4)

                    return true;

            }

        }

        else if(this.checkIfStackableExist(item\_num) != -1)

            return true;

        return false;

    }

    //1:可合成 -1:材料夠沒空間 0:材料不夠

    this.checkIfSynthesisAvailable = function(material){

        if(this.checkIfMaterialEnough(material)){

            if(this.itemList.length < 17)

                return 1;

            if(this.checkIfSpaceEnoughAfterSynthesis(material))

                return 1;

            else

                return -1;

        }else{

            return 0;

        }

    }

    this.checkIfSpaceEnoughAfterSynthesis = function(material){

        var check = false;

        for(var i = 0;i < material.length-1;i++){

            for(var j = 0;j < this.itemList.length;j++){

                if(this.itemList[j].item\_num == material[i].item && this.itemList[j].amount == material[i].amount)

                    check = true;

            }

        }

        return check;

    }

    this.checkIfMaterialEnough = function(material){

        var check = false;

        for(var i = 0;i < material.length-1;i++){

            check = false;

            for(var j = 0;j < this.itemList.length;j++){

                if(this.itemList[j].item\_num == material[i].item && this.itemList[j].amount >= material[i].amount)

                    check = true;

            }

            if(!check)

                return false;

        }

        return true;

    }

    this.checkIfStackableExist = function(item\_num){

        if(this.stackableList.indexOf(item\_num) != -1){

            for(var i = 0;i < this.itemList.length;i++){

                if(this.itemList[i].item\_num == item\_num)

                    return i;

            }

        }

        return -1;

    }

    this.getItem = function(index){

        return this.itemList[index];

    }

    this.getItemListLength = function(){

        return this.itemList.length;

    }

    this.update = function(material){

        for(var i = 0;i < material.length-1;i++){

            for(var j = 0;j <　this.itemList.length;j++){

                if(material[i].item == this.itemList[j].item\_num)

                    this.itemList[j].amount -= material[i].amount;

            }

        }

        this.arrayRemoveEmpty();

    }

    this.updateByEat = function(index){

        this.itemList[index].amount -= 1;

        if(this.itemList[index].amount == 0)

            this.arrayRemoveByIndex(index);

    }

    this.arrayRemoveByIndex = function(i){

        this.itemList.splice(i, 1);

    }

    this.arrayRemoveEmpty = function() {

        var i = 0;

        while(i < this.itemList.length)

            if (this.itemList[i].amount == 0)

                this.itemList.splice(i, 1);

            else

                i++;

    }

    //1:小花 2:蜘蛛網 3:石頭 4:樹枝 5:偉凱的作業簿 6:草 7:木頭 8:燧石 9:黃金 10:豬皮

    //11:蜂刺 12:雪球 13:繩索 14:露水 15:十字鎬 16:斧頭 17:釣魚竿 18:鏟子 19:黃金斧頭 20:黃金鏟子

    //21:黃金十字鎬 22:頭盔 23:草製盔甲 24:木製盔甲 25:長矛 26:吹箭 27:國王法杖 28:空間法杖 29:火法杖 30:冰法杖

    //31:黃金提燈 32:火把 33:帳篷 34:篝火 35:冰塊 36:漿果叢

    this.draw = function(ctx){

        for(var j = 0; j < 17; j++){

            this.backpack.position = this.backpackPosition[j];

            this.backpack.draw(ctx);

        }

        for(var i = 0; i < this.itemList.length; i++){

            this.itemList[i].position = this.objectPosition[i];

            this.itemList[i].draw(ctx);

            ctx.font = "25px Arial";

            ctx.fillStyle = "white";

            ctx.strokeStyle = 'blue';

            ctx.textAlign = 'center';

            ctx.lineWidth = 2.5;

            if(this.stackableList.indexOf(this.itemList[i].item\_num) != -1){

                ctx.fillText(this.itemList[i].amount.toString(), this.backpackPosition[i].x, this.backpackPosition[i].y+20);

                ctx.strokeText(this.itemList[i].amount.toString(), this.backpackPosition[i].x, this.backpackPosition[i].y+20);

            }else{

                ctx.fillText(this.itemList[i].durability.toString()+"%", this.backpackPosition[i].x, this.backpackPosition[i].y+20);

                ctx.strokeText(this.itemList[i].durability.toString()+"%", this.backpackPosition[i].x, this.backpackPosition[i].y+20);

            }

        }

    }

};

1. bat\_wing.js

var Bat\_wing = function() {

    this.item\_bat\_wing = new Framework.Sprite(define.materialPath + 'Batilisk\_Wing.png');

    this.item\_bat\_wing.scale = 0.8;

    this.item\_num = 49;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_bat\_wing, 1);

};

Bat\_wing.prototype = new Food\_base();

1. bombMan.js

//由於JS尚未支援Class(ECMAScript 6以後, 宣稱會支援)

//目前Class寫法都是以function的方式

//只要是this.XXX皆會是Public的property

var BombMan = function(file, options) {

    this.url = file;

    this.item\_bush = new Framework.Sprite(define.materialPath + 'item\_bush.png');

    this.item\_bush.scale = 2;

    this.item\_bush.position = {x:13\*64,y:7\*64};

    this.hurt = new Framework.AnimationSprite({url:define.materialPath + 'Absorb.png', col:5, row:5, loop:false, speed:12});

    this.hurt.position = {x:13\*64,y:7\*64};

    this.hurt.scale = 2;

    this.transport = new Framework.AnimationSprite({url:define.skillAnimationPath + 'Cure3.png', col:5, row:6, loop:false, speed:12});

    this.transport.position = {x:11.5\*64,y:5\*64};

    this.transport.scale = 1;

    this.hideAnimation = new Player\_hide\_animation();

    this.get\_back\_number;

    this.sprite\_dead = new Framework.AnimationSprite({url:define.materialPath + 'Actor\_dead.png', col:3 , row:1 , loop:false , speed:4});

    this.sprite\_dead.scale = 1.4;

    this.player\_state = "";

    //AnimationSprite當圖片是一整張圖片(連續圖), 而非Array時一定要給col, row三個(url是一定要的)

    this.sprite = new Framework.AnimationSprite({url:this.url, col:3 , row:4 , loop:true , speed:12});

    this.sprite.scale = 1.4;

    this.sprite.index = 1;

    var PIXEL\_CONST = 64;

    this.canvasPosition = {x:0, y:0};

    this.spritePosition = {x:0, y:0};

    this.constants = new Constants();

    this.StepMovedCallBack = [];

    this.maxBombNum = 1;

    this.bombNum = 0;

    this.pic\_count = 0;

    this.isWalking = false;

    var m\_bombMan = this;

    this.playerDirection = this.constants.DirectionEnum.DOWN;

    this.equipmentBar = new EquipmentBar();

    this.backpack = new Backpack();

    this.hide = false;

    this.gameClear = false;

    this.mode = "";

    this.beCaught = false;

    //一個格子 = 20

    this.character\_descruption\_point = [];

    this.character\_descruption\_point[0] = 0\*20;   //"生命";

    this.character\_descruption\_point[1] = 0\*20;   //"魔力";

    this.character\_descruption\_point[2] = 0\*20;   //"物功";

    this.character\_descruption\_point[3] = 0\*20;   //"魔攻";

    this.character\_descruption\_point[4] = 0\*20;   //"弓攻";

    this.character\_descruption\_point[5] = 0\*20;   //"力量";

    this.character\_descruption\_point[6] = 0\*20;  //"智力";

    this.character\_descruption\_point[7] = 0\*20;   //"防禦";

    this.character\_descruption\_point[8] = 0\*20;  //"技巧";

    //角色格子轉成數值

    this.character\_descruption\_total\_point = [];

    this.character\_descruption\_total\_point[0] = 0;

    this.character\_descruption\_total\_point[1] = 0;

    this.character\_descruption\_total\_point[2] = 0;

    this.character\_descruption\_total\_point[3] = 0;

    this.character\_descruption\_total\_point[4] = 0;

    this.character\_descruption\_total\_point[5] = 0;

    this.character\_descruption\_total\_point[6] = 0;

    this.character\_descruption\_total\_point[7] = 0;

    this.character\_descruption\_total\_point[8] = 0;

    this.experience = 0;

    this.levelup\_experience = 4;

    this.level = 1;

    this.is\_levelup = false;

    this.capabilityt\_point = 0;

    this.level\_up\_animation = new Level\_up\_animation();

    this.totalDefense = 10;

    this.plantIndex = -1;

    this.hunger\_total\_point = 200;

    this.hunger\_current\_point = 100;

    this.handEquipmentId;

    this.init = function(){

        this.player\_state = "alive";

        this.decreaseHunger();

    }

    this.attack = function(attackMode){

        attackMode.attack();

    }

    this.hidePlayer = function(){

        this.equipmentBar.equipmentList[2].reduceDurability();

        this.hide = true;

        setTimeout(()=>{

            this.hide = false;

        },10000);

    }

    this.transportPlayer = function(){

        this.equipmentBar.equipmentList[2].reduceDurability();

        this.transport.start({from:0,to:6,loop:false});

    }

    this.decreaseHunger = function(){

        var hungerInterval = setInterval(()=>{

            this.hunger\_current\_point -= 1;

            if(this.gameClear){

                clearInterval(hungerInterval);

            }

            else if(this.hunger\_current\_point <= 0)

            {

                clearInterval(hungerInterval);

                this.decreaseHealth();

            }

        }, 1500);

    }

    this.decreaseHealth = function(){

        var healthInterval = setInterval(()=>{

            this.character\_descruption\_point[0] -= 4;

            if(this.gameClear){

                clearInterval(healthInterval);

            }

            else if(this.hunger\_current\_point > 0 && !(this.character\_descruption\_point[0] <= 0)){

                clearInterval(healthInterval);

                this.decreaseHunger();

            }else if(this.character\_descruption\_point[0] <= 0){

                clearInterval(healthInterval);

            }

        }, 1500);

    }

    this.getExperience= function(experience){

        this.experience += experience;

        if(this.experience >=this.levelup\_experience){

            this.experience -= this.levelup\_experience

            this.levelup\_experience \*= 2;

            this.level ++;

            this.capabilityt\_point ++;

            this.level\_up\_animation.start();

            this.audio = new Framework.Audio({

                kick: {

                    mp3: define.musicPath + 'levelup.mp3',

                },

                song1:{

                    mp3: define.musicPath + 'easy.mp3',

                },

                die\_scream:{

                    mp3: define.musicPath + '女慘叫.mp3',

                },

                monster\_attack:{

                    mp3: define.musicPath + 'monster\_attack.mp3',

            }});

            this.audio.play({name: 'kick', loop: false});

        }

    }

    this.gethurt = function(attack\_point){

        var damage = attack\_point - (this.character\_descruption\_total\_point[7]/2);

        if(this.equipmentBar.getEquipment(0) != null)

            this.equipmentBar.getEquipment(0).reduceDurability();

        if(this.equipmentBar.getEquipment(1) != null)

            this.equipmentBar.getEquipment(1).reduceDurability();

        if(damage > 0){

            this.hurt.start({ from: 17, to: 18, loop: false});

            this.character\_descruption\_point[0] -= damage;

            if(this.character\_descruption\_point[0]<0)

                this.character\_descruption\_point[0] = 0;

        }

    }

    this.getBackPack = function(){

        return this.backpack;

    }

    this.getHeadDeffensePointEquipment = function(){

        var bodyEquipment = this.equipmentBar.getEquipment(0);

        if(bodyEquipment != null)

            return this.equipmentBar.getEquipment(0).deffense\_point;

        else

            return 0;

    }

    this.getBodyDeffensePointEquipment = function(){

        var bodyEquipment = this.equipmentBar.getEquipment(1);

        if(bodyEquipment != null)

            return this.equipmentBar.getEquipment(1).deffense\_point;

        else

            return 0;

    }

    this.getHandAttackPointEquipment = function(){

        var handEquipment = this.equipmentBar.getEquipment(2);

        if(handEquipment != null)

            return this.equipmentBar.getEquipment(2).attack\_point;

        else

            return 0;

    }

    this.getHandMagicAttackPointEquipment = function(){

        var handEquipment = this.equipmentBar.getEquipment(2);

        if(handEquipment != null)

            return this.equipmentBar.getEquipment(2).magic\_attack\_point;

        else

            return 0;

    }

    this.getHandArrorAttackPointEquipment = function(){

        var handEquipment = this.equipmentBar.getEquipment(2);

        if(handEquipment != null)

            return this.equipmentBar.getEquipment(2).arror\_attack\_point;

        else

            return 0;

    }

    this.getHandEquipment = function(){

        var handEquipment = this.equipmentBar.getEquipment(2);

        if(handEquipment != null){

            this.handEquipmentId = handEquipment.item\_num;

            if(handEquipment.item\_num == 32)

                this.mode = "light";

            else if(handEquipment.item\_num == 16 || handEquipment.item\_num == 19)

                this.mode = "cut\_tree";

            else if(handEquipment.item\_num == 15 || handEquipment.item\_num == 21)

                this.mode = "rock\_dig";

            else if(handEquipment.item\_num == 18 || handEquipment.item\_num == 20)

                this.mode = "plant\_dig";

            else if(handEquipment.item\_num == 29 || handEquipment.item\_num == 30)

                this.mode = "magic";

            else if(handEquipment.item\_num == 28)

                this.mode = "space";

            else if(handEquipment.item\_num == 25)

                this.mode = "spear";

            else if(handEquipment.item\_num == 26)

                this.mode = "arror";

            else if(handEquipment.item\_num == 17)

                this.mode = "fishing";

            else if(handEquipment.item\_num == 27)

                this.mode = "hide";

            else

                this.mode = "";

        }else{

            this.handEquipmentId = null;

            this.mode = "";

        }

    }

    this.dieEvent = function(position){

        this.player\_state = "dead";

        this.character\_descruption\_point[0] = 0;

        this.character\_descruption\_total\_point[0] = 0;

        this.sprite\_dead.position = {x: position.x\*64, y: position.y\*64};

        this.sprite\_dead.start({ from: 0 , to: 2, loop: true});

    }

    this.walk = function(moveStep){

        if(this.isWalking === false){

            if(moveStep.x > 0)

                this.playerDirection = this.constants.DirectionEnum.RIGHT;

            else if(moveStep.x <0)

                this.playerDirection = this.constants.DirectionEnum.LEFT;

            if(moveStep.y > 0)

                this.playerDirection = this.constants.DirectionEnum.DOWN;

            else if(moveStep.y < 0)

                this.playerDirection = this.constants.DirectionEnum.UP;

            this.isWalking = true;

            this.sprite.start({ from: this.playerDirection \* 3, to: this.playerDirection \* 3 + 2, loop: true});

        }

    }

    this.die = function(){

        Framework.Game.goToLevel('gameOver');

    }

    this.walkEnd = function(){

        this.sprite.stop();

    }

    this.setCapibility = function(capability){

        for(var i=0;i<9;i++)

            this.character\_descruption\_point[i] = capability[i]\*20;

        this.character\_descruption\_point[0] = capability[5]\*20;

    }

    this.capibility = function(){

        for(var i=0;i<9;i++)

            this.character\_descruption\_total\_point[i] = this.character\_descruption\_point[i];

        this.character\_descruption\_total\_point[2] += this.getHandAttackPointEquipment();

        this.character\_descruption\_total\_point[3] += this.getHandMagicAttackPointEquipment();

        this.character\_descruption\_total\_point[4] += this.getHandArrorAttackPointEquipment();

        this.character\_descruption\_total\_point[7] += this.getBodyDeffensePointEquipment()+this.getHeadDeffensePointEquipment();

    }

    this.update = function(){

        //更新升級動畫

        this.level\_up\_animation.update();

        this.hideAnimation.update();

        this.hurt.update();

        //更新角色血量(飢餓狀態)  this.characterStatus.currentHunger是數值,要轉成格子

        this.capibility();

        if(this.isWalking){

            this.isWalking = false;

            this.sprite.index = this.playerDirection \* 3 + 1;

            for(var i=0; i<this.StepMovedCallBack.length; i++){

                this.StepMovedCallBack[i];

            }

        }

        this.equipmentBar.update();

        this.getHandEquipment();

        this.sprite\_dead.update();

        this.sprite.update();

        this.transport.update();

    }

    this.draw = function(ctx){

        if(this.hideAnimation.hideAnimation.\_start)

            this.hideAnimation.draw(ctx);

        if(this.level\_up\_animation.level\_up\_animation.\_start){

            this.level\_up\_animation.draw(ctx);

        }

        if(this.transport.\_start)

            this.transport.draw(ctx);

        this.sprite.position = {x: this.spritePosition.x, y: this.spritePosition.y};

        this.equipmentBar.draw(ctx);

        this.backpack.draw(ctx);

        if(this.player\_state == "alive" && !this.hide && !this.transport.\_start)

            this.sprite.draw(ctx);

        else if(this.player\_state == "dead")

            this.sprite\_dead.draw(ctx);

        else if(this.hide)

            this.item\_bush.draw(ctx);

    }

    this.equipFromBackpack = function(indexForBackpack, indexForEquipment){

        var obj = this.backpack.getItem(indexForBackpack);

        if(obj.item\_num == 32){

            obj.inEquipmentbar = true;

            obj.reduceDurability();

        }

        var equipment\_obj = this.equipmentBar.getEquipment(indexForEquipment);

        this.backpack.arrayRemoveByIndex(indexForBackpack);

        this.equipmentBar.setEquipment(obj, indexForEquipment);

        if(equipment\_obj != null){

            if(equipment\_obj.item\_num == 32)

                equipment\_obj.inEquipmentbar = false;

            this.backpack.addItemByObject(equipment\_obj);

        }

    }

    this.clickInBackPack = function(index){

        this.plantIndex = -1;

        var obj = this.backpack.getItem(index);

        if (obj.type == "equipment"){

            switch(obj.place){

                case "head":

                    this.equipFromBackpack(index, 0);

                    break;

                case "body":

                    this.equipFromBackpack(index, 1);

                    break;

                case "hand":

                    this.equipFromBackpack(index, 2);

                    break;

            }

        }

        if (obj.type == "food"){

            this.increaseStatusByEat(obj.hungerAddition, obj.healthAddition);

            this.backpack.updateByEat(index);

        }

        if(obj.type == "plant")

            this.plantIndex = index;

    }

    this.increaseStatusByEat = function(hunger, health){

        if(this.character\_descruption\_point[0]  + health >= this.character\_descruption\_point[5])

            this.character\_descruption\_point[0] = this.character\_descruption\_point[5];

        else

            this.character\_descruption\_point[0] += health;

        if(this.hunger\_current\_point + hunger >= this.hunger\_total\_point)

            this.hunger\_current\_point = this.hunger\_total\_point;

        else

            this.hunger\_current\_point += hunger;

    }

    this.removeEquipment = function(index){

        if(this.backpack.getItemListLength() < 17 && this.equipmentBar.getEquipment(index) != null){

            if(this.equipmentBar.getEquipment(index).item\_num == 32)

                this.equipmentBar.getEquipment(index).inEquipmentbar = false;

            this.backpack.addItemByObject(this.equipmentBar.getEquipment(index));

            this.equipmentBar.setEquipment(null, index);

        }

    }

    this.mousemove = function(e){

        var index = this.getBackPackIndex(e);

        var equipmentIndex = this.getEquipmentIndex(e);

        this.backpack.selectedIndex = index;

        this.equipmentBar.selectedIndex = equipmentIndex;

    }

    this.isChangeCapability = function(which\_capability){

        this.character\_descruption\_point[which\_capability] += 20;

        this.capabilityt\_point--;

    }

    this.characterAbilityClick = function(e){

        if(e.x >= 470 && e.x <=510){

            if(e.y >=540 && e.y<= 560)

                this.isChangeCapability(2);

            else if(e.y >=622 && e.y<= 642)

                this.isChangeCapability(3);

            else if(e.y >=700 && e.y<= 720)

                this.isChangeCapability(4);

        }else if(e.x >= 822 && e.x <=861){

            if(e.y >=382 && e.y<= 415)

                this.isChangeCapability(5);

            else if(e.y >=461 && e.y<= 502)

                this.isChangeCapability(6);

            else if(e.y >=622 && e.y<= 661)

                this.isChangeCapability(8);

        }

    }

    this.click = function(e){

        var index = this.getBackPackIndex(e);

        var equipmentIndex = this.getEquipmentIndex(e);

        if(index != -1)

            this.clickInBackPack(index);

        if(equipmentIndex != -1)

            this.removeEquipment(equipmentIndex);

    }

    this.getBackPackIndex = function(e){

        var index = -1;

        if(e.y >= 800 && e.y <=860){

            if(e.x >= 290 && e.x < 350)

                index = 0;

            else if(e.x >= 350 && e.x < 410)

                index = 1;

            else if(e.x >= 410 && e.x < 480)

                index = 2;

            else if(e.x >= 480 && e.x < 540)

                index = 3;

            else if(e.x >= 540 && e.x < 605)

                index = 4;

            else if(e.x >= 605 && e.x < 672)

                index = 5;

            else if(e.x >= 672 && e.x < 732)

                index = 6;

            else if(e.x >= 732 && e.x < 800)

                index = 7;

            else if(e.x >= 800 && e.x < 860)

                index = 8;

            else if(e.x >= 860 && e.x < 924)

                index = 9;

            else if(e.x >= 924 && e.x < 990)

                index = 10;

            else if(e.x >= 990 && e.x < 1053)

                index = 11;

            else if(e.x >= 1053 && e.x < 1118)

                index = 12;

            else if(e.x >= 1118 && e.x < 1183)

                index = 13;

            else if(e.x >= 1183 && e.x < 1247)

                index = 14;

            else if(e.x >= 1247 && e.x < 1310)

                index = 15;

            else if(e.x >= 1310 && e.x < 1370)

                index = 16;

        }

        return index;

    }

    this.getEquipmentIndex = function(e){

        var equipmentIndex = -1;

        if(e.x >= 1443 && e.x <= 1498){

            if(e.y >= 547 && e.y <= 606)

                equipmentIndex = 0;

            if(e.y >= 606 && e.y <= 667)

                equipmentIndex = 1;

            if(e.y >= 667 && e.y <= 733)

                equipmentIndex = 2;

        }

        return equipmentIndex;

    }

};

Object.defineProperty(BombMan.prototype, 'position', {

    get: function() {

        return this.canvasPosition;

    },

    set: function(newValue) {

        this.canvasPosition = newValue;

        this.spritePosition = {x:this.canvasPosition.x \* 64, y: this.canvasPosition.y \* 64};

    }

});

1. building\_remance.js

var Building\_remance = function(number) {

    this.building\_remance = [];

    this.building\_remance.push(new Framework.Sprite(define.builldingPath + '遺跡1.png'));

    this.building\_remance[0].scale = 2

    this.building\_remance.push(new Framework.Sprite(define.builldingPath + '遺跡2.png'));

    this.building\_remance[1].scale = 2

    this.building\_remance.push(new Framework.Sprite(define.builldingPath + '遺跡3.png'));

    this.building\_remance[2].scale = 2

    this.isRegenerate = false;

    this.mapPosition = {x:0, y:0};

    this.item\_num = -5;

    this.type = "build";

    this.amount = 1;

    this.item\_can\_be\_picked = false;

    this.draw = function(ctx){

        this.building\_remance[number].position = this.mapPosition;

        this.building\_remance[number].draw(ctx);

    }

};

Object.defineProperty(Building\_remance.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.mapPosition = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. character\_description.js

var Character\_description = function() {

    this.back\_ground\_picture = new Framework.Sprite(define.characterDescriptionPath + 'character\_descript\_scroll.png');

    this.character\_image = new Framework.Sprite(define.characterDescriptionPath + 'character\_image.png');

    this.experience\_chart = new Framework.Sprite(define.characterDescriptionPath + 'experience\_chart.png');

    this.character\_level = 1;

    this.character\_descruption = [];

    this.character\_descruption[0] = new Framework.Sprite(define.characterDescriptionPath + 'health\_point.png');

    this.character\_descruption[1] = new Framework.Sprite(define.characterDescriptionPath + 'magic\_point.png');

    this.character\_descruption[2] = new Framework.Sprite(define.characterDescriptionPath + 'physical\_attack.png');

    this.character\_descruption[3] = new Framework.Sprite(define.characterDescriptionPath + 'magic\_attack.png');

    this.character\_descruption[4] = new Framework.Sprite(define.characterDescriptionPath + 'arrow\_attack.png');

    this.character\_descruption[5] = new Framework.Sprite(define.characterDescriptionPath + 'power.png');

    this.character\_descruption[6] = new Framework.Sprite(define.characterDescriptionPath + 'intelligence.png');

    this.character\_descruption[7] = new Framework.Sprite(define.characterDescriptionPath + 'defense.png');

    this.character\_descruption[8] = new Framework.Sprite(define.characterDescriptionPath + 'skill.png');

    this.push\_button = new Framework.Sprite(define.characterDescriptionPath + 'push\_button.png');

    this.character\_descruption\_point = [];

    this.character\_descruption\_point[0] = 0;

    this.character\_descruption\_point[1] = 0;

    this.character\_descruption\_point[2] = 0;

    this.character\_descruption\_point[3] = 0;

    this.character\_descruption\_point[4] = 0;

    this.character\_descruption\_point[5] = 0;

    this.character\_descruption\_point[6] = 0;

    this.character\_descruption\_point[7] = 0;

    this.character\_descruption\_point[8] = 0;

    this.character\_descruption\_text = [];

    this.character\_descruption\_text[0] = "生命";

    this.character\_descruption\_text[1] = "魔力";

    this.character\_descruption\_text[2] = "物功";

    this.character\_descruption\_text[3] = "魔攻";

    this.character\_descruption\_text[4] = "弓攻";

    this.character\_descruption\_text[5] = "力量";

    this.character\_descruption\_text[6] = "智力";

    this.character\_descruption\_text[7] = "防禦";

    this.character\_descruption\_text[8] = "技巧";

    this.push\_button.scale = 0.5;

    this.back\_ground\_picture.scale = 3;

    this.character\_image.scale = 0.75;

    this.experience\_chart.scale = 4;

    this.capabilityt\_point = 0;

    this.experience\_chart.position = {x: 14\*64, y: 2\*64};

    this.back\_ground\_picture.position = {x: 13\*64, y: 7\*64};

    this.character\_image.position = {x:11\*64, y:4\*64};

    this.is\_character\_description\_open = false;

    this.character\_current\_experience = 0;

    this.character\_levelup\_experience = 10;

    for(var i=0;i<9;i++){

        this.character\_descruption[i].scale = 0.6;

        if(i<5){

            this.character\_descruption[i].position = {x: 8.5\*64, y:7\*64+i\*80-48};

        }else{

            this.character\_descruption[i].position = {x: 14\*64, y:i\*80};

        }

    }

    this.draw = function(ctx){

        if(this.is\_character\_description\_open){

                this.back\_ground\_picture.draw(ctx);

            this.character\_image.draw(ctx);

            for(var i=0;i<9;i++){

                this.character\_descruption[i].draw(ctx);

                ctx.font = "bold 24px serif";

                ctx.fillStyle = "black";

                ctx.lineWidth = 2.5;

                ctx.fillText(this.character\_descruption\_text[i], this.character\_descruption[i].position.x+55,  this.character\_descruption[i].position.y+5);

                for(var j=0;j<this.character\_descruption\_point[i];j++){

                    ctx.beginPath();

                    ctx.rect(this.character\_descruption[i].position.x+95 + j\*13, this.character\_descruption[i].position.y-10, 10, 15);

                    if(this.character\_descruption\_point[i]<6){

                        ctx.fillStyle = "Red";

                    }else if(this.character\_descruption\_point[i]<11){

                        ctx.fillStyle = "yellow";

                    }else{

                        ctx.fillStyle = "Green";

                    }

                    ctx.fill();

                    ctx.closePath();

                }

            }

            ctx.font = "bold 48px serif";

            ctx.fillStyle = "black";

            ctx.lineWidth = 2.5;

            ctx.fillText("Lv " + this.character\_level, 10\*64+20, 2\*64);

            var experience\_scale;

            if(this.experience == 0)

                experience\_scale = -1;

            else

                experience\_scale = 15/(this.character\_levelup\_experience/this.experience);

            for(var i=0;i<=experience\_scale;i++){

                ctx.beginPath();

                ctx.rect(729+i\*23, 98, 20, 30);

                ctx.fillStyle = "blue";

                ctx.fill();

                ctx.closePath();

            }

            this.experience\_chart.draw(ctx);

            ctx.font = "bold 24px serif";

            ctx.fillStyle = "black";

            ctx.lineWidth = 2.5;

            ctx.fillText("( " + this.experience+" / "+ this.character\_levelup\_experience +")", 1135, 2\*64-5);

            ctx.font = "bold 24px serif";

            ctx.fillStyle = "black";

            ctx.lineWidth = 2.5;

            ctx.fillText("能力值點數: "+this.capabilityt\_point, 955, 315);

            if(this.capabilityt\_point!=0){

                for(var i=2;i<9;i++){

                    if(i==7)

                        continue;

                    this.push\_button.position = {x:this.character\_descruption[i].position.x-56, y:this.character\_descruption[i].position.y};

                    this.push\_button.draw(ctx);

                }

            }

        }else{

            ctx.beginPath();

            ctx.rect(449, 31, 30, 768);

            ctx.rect(1183, 31, 30, 768);

            ctx.rect(477, 32, 710, 65);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

        }

    }

    this.isChangeCapability = function(which\_capability){

        this.character\_descruption\_point[which\_capability] ++;

        this.capabilityt\_point--;

        this.draw(Framework.Game.\_context);

    }

    this.update = function(player){

        this.experience = player.experience;

        this.character\_level = player.level;

        this.character\_levelup\_experience = player.levelup\_experience;

        this.capabilityt\_point = player.capabilityt\_point;

        this.character\_descruption\_point[0] = player.character\_descruption\_total\_point[0]/20;

        this.character\_descruption\_point[1] = player.character\_descruption\_total\_point[1]/20;

        this.character\_descruption\_point[2] = player.character\_descruption\_total\_point[2]/20;

        this.character\_descruption\_point[3] = player.character\_descruption\_total\_point[3]/20;

        this.character\_descruption\_point[4] = player.character\_descruption\_total\_point[4]/20;

        this.character\_descruption\_point[5] = player.character\_descruption\_total\_point[5]/20;

        this.character\_descruption\_point[6] = player.character\_descruption\_total\_point[6]/20;

        this.character\_descruption\_point[7] = player.character\_descruption\_total\_point[7]/20;

        this.character\_descruption\_point[8] = player.character\_descruption\_total\_point[8]/20;

    }

};

1. clock.js

var Clock = function() {

    this.clock = [];

    this.timelevel = new Framework.Sprite(define.materialPath + 'timelevel.png');

    this.day\_board = new Framework.Sprite(define.materialPath + 'Home\_Sign.png');

    this.day\_board.scale = 3;

    this.day\_board.position = {x:22.2\*64, y:3.7\*64}

    this.music\_stop = false;

    this.timelevel.scale = 4;

    this.timelevel.position = {x: 23.75\*64, y: 2\*64};

    for(var i=0;i<3;i++){

        var clock\_child = new Framework.Sprite(define.materialPath + 'clock' + i + '.png');

        clock\_child.scale = 2;

        clock\_child.position = {x: 22\*64, y: 2\*64};

        this.clock.push(clock\_child);

    }

    this.status = 0;

    this.currentTime = 192;

    this.regeneration\_time = 200;

    this.day = 1;

    this.init = function(){

        var audio = new Framework.Audio({

            morning:{

                mp3: define.musicPath + 'Hot\_Heat.mp3'

            }, night:{

                mp3: define.musicPath + 'night.mp3'

            }, afternoon:{

                mp3: define.musicPath + '遊戲王經典配樂.mp3'

            }

        });

        audio.play({name: 'morning', loop: true});

        this.decrease();

    }

    this.decrease = function(){

        if(!this.music\_stop){

            setTimeout(()=>{

                this.currentTime -= 1;

                if(this.currentTime <= 0){

                    this.currentTime = 192;

                    var audio = new Framework.Audio({

                        morning:{

                            mp3: define.musicPath + 'Hot\_Heat.mp3'

                        }, night:{

                            mp3: define.musicPath + 'night.mp3'

                        }, afternoon:{

                            mp3: define.musicPath + '遊戲王經典配樂.mp3'

                        }

                    });

                    if(this.status == 0){

                        audio.stopAll();

                        audio.play({name: 'afternoon', loop: true});

                    }else if(this.status == 1){

                        audio.stopAll();

                        audio.play({name: 'night', loop: true});

                    }else{

                        audio.stopAll();

                        audio.play({name: 'morning', loop: true});

                    }

                    this.status ++;

                    if(this.status >=3){

                        this.status = 0;

                        this.day += 1;

                    }

                }

                this.decrease();

            }, this.regeneration\_time);

        }else{

            audio.stopAll();

        }

    }

    this.stopMusic = function(music\_stop){

        this.music\_stop = music\_stop;

    }

    this.draw = function(ctx){

        ctx.beginPath();

        ctx.rect(1290, 20, 300, 290);

        ctx.fillStyle = "#BEBEBE";

        ctx.fill();

        this.day\_board.draw(ctx);

        ctx.beginPath();

        ctx.rect(23.6\*64, 0.5\*64, 16, 192);

        ctx.fillStyle = "yellow";

        ctx.fill();

        ctx.beginPath();

        ctx.rect(23.6\*64, 0.5\*64, 16, this.currentTime);

        ctx.fillStyle = "black";

        ctx.fill();

        this.timelevel.draw(ctx);

        this.clock[this.status].draw(ctx);

        ctx.font = '25pt Times New Roman';

        ctx.fillStyle = 'yellow';

        ctx.strokeStyle = 'black';

        ctx.lineWidth = 3;

        ctx.strokeText("Day: " + this.day, 22.2\*64, 4.1\*64+20);

    }

};

1. constants.js

var Constants = function() {

    this.DirectionEnum = {

        DOWN : 0,

        LEFT : 1,

        RIGHT : 2,

        UP : 3

    };

    this.Direction = {

        DOWN : {x:0, y:1},

        LEFT : {x:-1,y:0},

        RIGHT : {x:1,y:0},

        UP : {x:0,y:-1}

    };

};

1. creation\_blood\_status.js

var Creation\_blood\_status = function() {

    this.monsterarray = [];

    this.character\_tocan = new Framework.Sprite(define.materialPath + 'character\_tocan.png');

    this.character\_tocan.scale = 0.35;

    this.character\_tocan.position = {x: 64\*3+64, y:64\*3};

    this.character\_blood\_chart = new Framework.Sprite(define.materialPath + 'blood\_chart.png');

    this.character\_blood\_chart.scale = 1.75;

    this.character\_blood\_chart.position = {x: 64\*6, y: 64\*3-20}

    this.character\_magic\_chart = new Framework.Sprite(define.materialPath + 'blood\_chart.png');

    this.character\_magic\_chart.scale = 1.75;

    this.character\_magic\_chart.position = {x: 64\*6, y: 64\*3+25-20}

    this.character\_hunger\_chart = new Framework.Sprite(define.materialPath + 'blood\_chart.png');

    this.character\_hunger\_chart.scale = 1.75;

    this.character\_hunger\_chart.position = {x: 64\*6, y: 64\*3+50-20}

    this.character\_blood\_ratio = 1;

    this.character\_magic\_ratio = 1;

    this.character\_hunger\_ratio = 1;

    this.blood\_chart = new Framework.Sprite(define.materialPath + 'blood\_chart.png');

    this.blood\_chart.scale = 1.5;

    this.blood\_ratio = [];

    this.monster\_tocan;

    this.init = function(player){

        this.characterHungryUpdate(player);

        this.characterMagicUpdate(player);

        this.characterHungryUpdate(player);

    }

    this.draw = function(ctx){

        this.character\_tocan.draw(ctx);

        ctx.beginPath();

        ctx.rect(64\*5-10, 64\*3-12-20, 150\*this.character\_blood\_ratio, 12);

        ctx.fillStyle = "red";

        ctx.fill();

        ctx.closePath();

        this.character\_blood\_chart.draw(ctx);

        ctx.beginPath();

        ctx.rect(64\*5-10, 64\*3+12-20, 150\*this.character\_magic\_ratio, 12);

        ctx.fillStyle = "blue";

        ctx.fill();

        ctx.closePath();

        this.character\_magic\_chart.draw(ctx);

        ctx.beginPath();

        ctx.rect(64\*5-10+1, 64\*3+38-20, 150\*this.character\_hunger\_ratio, 12);

        ctx.fillStyle = "LightGoldenrodYellow";

        ctx.fill();

        ctx.closePath();

        this.character\_hunger\_chart.draw(ctx);

        if(this.monsterarray.length>8)

            this.monsterarray.length=8;

        if(this.monsterarray.length >0){

            for(var i=0;i<this.monsterarray.length;i++){

                this.monsterarray[i].draw(ctx);

                this.blood\_chart.position = this.monsterarray[i].position;

                this.blood\_chart.position.x += 64\*2 - 24;

                this.blood\_chart.position.y += 25 - 18;

                ctx.beginPath();

                ctx.rect(this.blood\_chart.position.x-64, this.blood\_chart.position.y-10, (this.blood\_ratio[i])\*128, 10);

                ctx.fillStyle = "red";

                ctx.fill();

                ctx.closePath();

                this.blood\_chart.draw(ctx);

            }

            this.monsterarray = [];

            this.blood\_ratio = [];

        }

    }

    this.characterHungryUpdate = function(player){

        this.character\_hunger\_ratio = player.hunger\_current\_point / player.hunger\_total\_point;

    }

    this.characterMagicUpdate = function(player){

        this.character\_magic\_ratio = player.character\_descruption\_point[1] / player.character\_descruption\_point[6];

    }

    this.characterBloodUpdate = function(player){

        this.character\_blood\_ratio = player.character\_descruption\_point[0] / player.character\_descruption\_point[5];

        if(this.character\_blood\_ratio<0)

            this.character\_blood\_ratio=0;

    }

    this.monsterUpdate = function(monsterarray){

        for(var i=0,j=0;i<monsterarray.length;i++){

            if(monsterarray[i].is\_start && monsterarray[i].health > 0){

                this.monster\_tocan = monsterarray[i].monster\_tocan;

                this.monster\_tocan.position = {x:65\*4+5, y:65\*(4+j)+10};

                this.blood\_ratio.push(monsterarray[i].health/monsterarray[i].maxHealth);

                this.monsterarray.push(this.monster\_tocan);

                j++;

            }

        }

    }

};

1. define.js

(function(window){

    var define ={}, mainPath = 'game\_sample/';

    Object.defineProperties(define, {

        'mainPath': {

            value: mainPath,

            writable: false

        },

        'jsPath': {

            value: mainPath + 'js/',

            writable: false

        },

        'musicPath': {

            value: mainPath + 'music/',

            writable: false

        },

        'imagePath': {

            value: mainPath + 'image/',

            writable: true

        },

        'builldingPath': {

            value: mainPath + 'building/',

            writable: true

        },

        'materialPath': {

            value: mainPath + 'material/',

            writable: true

        },

        'welcomimgPath': {

            value: mainPath + 'welcoming\_page\_pixel/',

            writable: true

        },

        'imageMorningPath': {

            value: mainPath + 'image\_morning/',

            writable: true

        },

        'imageAfternoonPath': {

            value: mainPath + 'image\_afternoon/',

            writable: true

        },

        'imageNightPath': {

            value: mainPath + 'image\_night/',

            writable: true

        },

        'skillAnimationPath': {

            value: mainPath + 'skill\_animation/',

            writable: true

        },

        'characterDescriptionPath': {

            value: mainPath + 'character\_description/',

            writable: true

        },

        'npcPath': {

            value: mainPath + 'npc/',

            writable: true

        }

    });

    window.define = define;

})(window)

1. drama0.js

var Drama0 = Framework.Class(Framework.Level , {

    initializeProgressResource: function() {

        this.loading = new Framework.Sprite(define.imagePath + 'loading.jpg');

        this.loading.position = {x: Framework.Game.getCanvasWidth() / 2 , y: Framework.Game.getCanvasHeight() / 2};

    },

    loadingProgress: function(ctx, requestInfo) {

        this.loading.draw(ctx);

        ctx.font ='90px Arial';

        ctx.textAlign = 'center';

        ctx.fillStyle = 'white';

        ctx.fillText(Math.round(requestInfo.percent) + '%' , ctx.canvas.width / 2 , ctx.canvas.height / 2 + 300);

    },

    load: function() {

        this.audio = new Framework.Audio({

            song1: {

                ogg: define.musicPath + '開場.mp3',

            }

        });

        this.audio.play({name: 'song1', loop: true});

        this.playerPositionOnMap = {x:0,y:0};

        this.npc\_event = new Npc\_event(this);

        this.npc\_event.trigger("主角", "drama0");

        this.npc\_event.talking();

    },

    draw:function(parentCtx){

        this.npc\_event.draw(parentCtx);

    },

    keydown:function(e, list){

        if(e.key === 'F11') {

            if(!this.isFullScreen) {

                Framework.Game.fullScreen();

                this.isFullScreen = true;

            } else {

                Framework.Game.exitFullScreen();

                this.isFullScreen = false;

            }

        }else if(e.key === 'Space'){

            this.npc\_event.talking();

            if( !this.npc\_event.taking\_is\_start){

                this.audio.stopAll();

                Framework.Game.goToLevel('level1');

            }

            this.draw(Framework.Game.\_context);

        }else if(e.key === 'Z'){

            this.npc\_event.amount -= 2;

            if( this.npc\_event.amount <= -1){

                this.npc\_event.amount = -1;

            }

            this.npc\_event.talking();

            this.draw(Framework.Game.\_context);

        }

    },

    click: function (e) {

    },

});

1. equipmentBar.js

var EquipmentBar = function(backpackList) {

    this.backpack = new Framework.Sprite(define.materialPath + 'backpack.png');

    this.backpack.scale = 2;

    this.backpack.position = {x: 23\*64, y: 8\*64};

    this.equipmentList = [null,null,null];

    this.selectedIndex = -1;

    this.draw = function(ctx){

        this.backpack.position = {x: 23\*64, y: 8\*64};

        for(var i = 0;i < 3;i++){

            this.backpack.position.y += 64;

            this.backpack.draw(ctx);

            if(this.equipmentList[i] != null){

                this.equipmentList[i].position = {x: 23, y: 9+i};

                this.equipmentList[i].draw(ctx);

                ctx.font = "25px Arial";

                ctx.fillStyle = "white";

                ctx.strokeStyle = 'blue';

                ctx.lineWidth = 2.5;

                ctx.fillText(this.equipmentList[i].durability.toString()+"%", this.equipmentList[i].position.x\*64, this.equipmentList[i].position.y\*64+15);

                ctx.strokeText(this.equipmentList[i].durability.toString()+"%", this.equipmentList[i].position.x\*64, this.equipmentList[i].position.y\*64+15);

            }

        }

    }

    this.getSelectedEquipment = function(){

        if(this.selectedIndex == -1 || this.equipmentList[this.selectedIndex] == null)

            return null;

        else{

            if(this.equipmentList[this.selectedIndex].item\_num == 32)

                this.equipmentList[this.selectedIndex].inEquipmentbar = false;

            return this.equipmentList[this.selectedIndex];

        }

    }

    this.dropSelectedEquipment = function(){

        if(this.selectedIndex != -1)

            this.equipmentList[this.selectedIndex] = null;

    }

    //0:head 1:body 2:hand

    this.getEquipment = function(index){

        return this.equipmentList[index];

    }

    this.setEquipment = function(obj, index){

        this.equipmentList[index] = obj;

    }

    this.update = function(){

        for(var i = 0;i < 3;i++){

            if(this.equipmentList[i] != null && this.equipmentList[i].durability <= 0){

                this.equipmentList[i] = null;

            }

        }

    }

};

1. fishing.js

var Fishing = function() {

    this.url = define.materialPath + 'water.PNG';

    this.url2 = define.materialPath + 'Balloon.PNG';

    this.fishing = new Framework.AnimationSprite({url:this.url, col:3 , row:4 , loop:true , speed:12});

    this.fishing.scale = 1;

    this.remindFishCatch = new Framework.AnimationSprite({url:this.url2, col:8 , row:15 , loop:false , speed:12});

    this.remindFishCatch.position = {x: (13)\*64, y: (6)\*64};

    this.remindFishCatch.scale = 0.8;

    this.is\_start = false;

    this.fishBeCaught = false;

    this.update = function(){

        this.fishing.update();

        this.remindFishCatch.update();

    }

    this.start = function(playerWalkDirection){

        this.is\_start = true;

        this.fishing.position = {x: (13+playerWalkDirection.x)\*64 -30, y: (7+playerWalkDirection.y)\*64 -25};

        this.fishing.start({ from: 0, to: 6, loop: true});

        this.waitForFishCatch();

    }

    this.waitForFishCatch = function(){

        var time = Math.floor(Math.random()\*8) + 3;

        setTimeout(()=>{

            if(this.is\_start)

                this.fishCatch();

        },time\*1000);

    }

    this.fishCatch = function(){

        this.fishBeCaught = true;

        this.remindFishCatch.start({ from: 0, to: 7, loop: true});

        setTimeout(()=>{

            this.fishBeCaught = false;

            this.remindFishCatch.stop();

            if(this.is\_start)

                this.waitForFishCatch();

        },1000);

    }

    this.stop = function(){

        this.fishing.stop();

        this.remindFishCatch.stop();

        this.is\_start = false;

    }

    this.draw = function(ctx){

        this.fishing.draw(ctx);

        if(this.fishBeCaught)

            this.remindFishCatch.draw(ctx);

    }

};

1. flying\_arror.js

var Flying\_arror = function(Direction,Position,Monster,Attack) {

    this.flying\_arror\_left = new Framework.Sprite(define.materialPath + 'arror\_left.png');

    this.flying\_arror\_left.scale = 2;

    this.flying\_arror\_right = new Framework.Sprite(define.materialPath + 'arror\_right.png');

    this.flying\_arror\_right.scale = 2;

    this.flying\_arror\_up = new Framework.Sprite(define.materialPath + 'arror\_up.png');

    this.flying\_arror\_up.scale = 2;

    this.flying\_arror\_down = new Framework.Sprite(define.materialPath + 'arror\_down.png');

    this.flying\_arror\_down.scale = 2;

    this.url = define.skillAnimationPath + 'Recovery4.png';

    this.flying\_arror = new Framework.AnimationSprite({url:this.url, col:5 , row:4 , loop:true , speed:16});

    this.flying\_arror.scale = 0.5;

    this.flying\_arror.start({ from: 0, to: 8, loop: false});

    this.directionNum;

    this.constants = new Constants();

    this.mapPosition = Position;

    this.flyTarget = Position;

    this.spritePosition = {x:Position.x \* 64,y:Position.y\*64};

    this.limitPosition = {x:Position.x + Direction.x\*5,y:Position.y + Direction.y\*5};

    this.direction = Direction;

    this.attackEnd = false;

    this.is\_start = false;

    this.init = function(){

        if(this.direction.x == this.constants.Direction.DOWN.x && this.direction.y == this.constants.Direction.DOWN.y)

            this.directionNum = 0;

        else if(this.direction.x == this.constants.Direction.LEFT.x && this.direction.y == this.constants.Direction.LEFT.y)

            this.directionNum = 1;

        else if(this.direction.x == this.constants.Direction.RIGHT.x && this.direction.y == this.constants.Direction.RIGHT.y)

            this.directionNum = 2;

        else

            this.directionNum = 3;

    }

    this.fly = function(){

        for(var i = 0;i < Monster.length;i++){

            if(Monster[i].is\_start){

                if(this.mapPosition.x == Monster[i].mapPosition.x && this.mapPosition.y == Monster[i].mapPosition.y){

                    Monster[i].health -= Attack;

                    Monster[i].getHurt();

                    this.attackEnd = true;

                    break;

                }

            }

        }

        if(this.mapPosition.x == this.limitPosition.x && this.mapPosition.y == this.limitPosition.y)

            this.attackEnd = true;

    }

    var flySpeed = 16;

    this.flyAlittle = function(){

        if(this.directionNum == 0)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y + flySpeed};

        else if(this.directionNum == 1)

            this.spritePosition = {x:this.spritePosition.x - flySpeed, y:this.spritePosition.y};

        else if(this.directionNum == 2)

            this.spritePosition = {x:this.spritePosition.x + flySpeed, y:this.spritePosition.y};

        else

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y - flySpeed};

    }

    this.update = function(){

        this.flying\_arror.update();

        if(this.flyTarget.x \* 64 === this.spritePosition.x && this.flyTarget.y \* 64 === this.spritePosition.y){

            this.mapPosition = this.flyTarget;

            this.flyTarget = {x:this.flyTarget.x+this.direction.x, y:this.flyTarget.y+this.direction.y};

            this.flyAlittle();

            this.fly();

        }else{

            this.flyAlittle();

            this.fly();

        }

    }

    this.check = function(){

        if( Math.abs((this.spritePosition.x - Position.x\*64)) <6\*64 &&　Math.abs((this.spritePosition.y - Position.y\*64)) <64\*6)

            this.is\_start = true;

        else

            this.is\_start = false;

    }

    this.draw = function(ctx){

        this.is\_start = false;

        this.check();

        if(this.is\_start){

            var xx = 13\*64 + this.spritePosition.x - Position.x\*64;

            var yy = 7\*64 + this.spritePosition.y - Position.y\*64;

            this.flying\_arror.position = {x: xx, y: yy};

            this.flying\_arror.draw(ctx);

            if(this.directionNum == 0){

                this.flying\_arror\_down.position = {x: xx, y: yy};

                this.flying\_arror\_down.draw(ctx);

            }

            else if(this.directionNum == 1){

                this.flying\_arror\_left.position = {x: xx, y: yy};

                this.flying\_arror\_left.draw(ctx);

            }

            else if(this.directionNum == 2){

                this.flying\_arror\_right.position = {x: xx, y: yy};

                this.flying\_arror\_right.draw(ctx);

            }

            else{

                this.flying\_arror\_up.position = {x: xx, y: yy};

                this.flying\_arror\_up.draw(ctx);

            }

        }

    }

};

Object.defineProperty(Flying\_arror.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.flying\_arror.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. food\_base.js

var Food\_base = function() {

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    //可疊加物件有amount 不可疊加有durability

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_can\_be\_picked = true;

    this.type = "food";

    this.init = function(food, amount){

        this.food = food;

        this.amount = amount;

    }

    this.draw = function(ctx){

        this.food.draw(ctx);

    }

};

Object.defineProperty(Food\_base.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.food.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. game\_object\_detail.js

var Game\_object\_detail = function() {

    this.back\_ground\_picture = new Framework.Sprite(define.materialPath + 'Floral.png');

    this.back\_ground\_picture.scale = 2.9;

    this.back\_ground\_picture.position = {x:145, y:720};

    //1:小花 2:蜘蛛網 3:石頭 4:樹枝 5:偉凱的作業簿 6:草 7:木頭 8:燧石 9:黃金 10:豬皮

    //11:蜂刺 12:雪球 13:繩索 14:露水 15:十字鎬 16:斧頭 17:釣魚竿 18:鏟子 19:黃金斧頭 20:黃金鏟子

    //21:黃金十字鎬 22:頭盔 23:草製盔甲 24:木製盔甲 25:長矛 26:吹箭 27:國王法杖 28:空間法杖 29:火法杖 30:冰法杖

    //31:黃金提燈 32:火把 33:帳篷 34:篝火 35:冰塊 36:漿果叢 37:採摘的花 38:採摘的草 39:漿果 40:挖的漿果叢(拔)

    //41:挖的花(拔) 42:挖的樹 43:挖的草(拔) 44:挖的漿果叢(未拔) 45:挖的花(未拔) 46:挖的草(未拔)

    this.description;

    this.init = function(){

        this.description =  [

                {key:1,name:"小花", des:"好可愛的一朵花上面有花蜜喔~"},

                {key:2,name:"蜘蛛網", des:"蜘蛛的家"},

                {key:3,name:"石頭", des:"萬用小石頭，是合成的好材料。"},

                {key:4,name:"樹枝", des:"小小一枝，不知到有什麼用處。"},

                {key:5,name:"偉凱的作業簿", des:"不好好寫功課，會被偉凱當掉喔~"},

                {key:6,name:"草", des:"草草草操操操!"},

                {key:7,name:"木頭", des:"木頭，不懂暗示的人"},

                {key:8,name:"燧石", des:"小小硬硬的"},

                {key:9,name:"黃金", des:"很稀有，要好好保存"},

                {key:10,name:"豬皮", des:"油到出水的一塊組織"},

                {key:11,name:"蜂刺", des:"從蜜蜂的屁屁上拔下來的，上面有一點毒液"},

                {key:12,name:"雪球", des:"白白的，做銼冰的好材料"},

                {key:13,name:"繩索", des:"非常的萬用"},

                {key:14,name:"露水", des:"跟明星花露水蠻像的"},

                {key:15,name:"十字鎬", des:"可以拿來挖礦"},

                {key:16,name:"斧頭", des:"不只可以砍樹，還可以砍人喔~"},

                {key:17,name:"釣魚竿", des:"釣釣釣"},

                {key:18,name:"鏟子", des:"挖哇哇"},

                {key:19,name:"黃金斧頭", des:"非常奢華，你捨得用嗎？"},

                {key:20,name:"黃金鏟子", des:"非常奢華，你捨得用嗎？"},

                {key:21,name:"黃金十字鎬", des:"非常奢華，你捨得用嗎？"},

                {key:22,name:"頭盔", des:"騎機車必帶的"},

                {key:23,name:"草製盔甲", des:"有點破爛的鎧甲"},

                {key:24,name:"木製盔甲", des:"還可以的鎧甲"},

                {key:25,name:"長矛", des:"獵殺怪物專用"},

                {key:26,name:"吹箭", des:"偷襲的好武器"},

                {key:27,name:"國王法杖", des:"消失吧！"},

                {key:28,name:"空間法杖", des:"移動吧！"},

                {key:29,name:"火法杖", des:"燃燒吧！"},

                {key:30,name:"冰法杖", des:"結凍吧！"},

                {key:31,name:"黃金提燈", des:"可以許願嗎？"},

                {key:32,name:"火把", des:"在夜晚派的上用場"},

                {key:33,name:"帳篷", des:"休息的好地方"},

                {key:34,name:"篝火", des:"烤肉囉"},

                {key:35,name:"冰塊", des:"好冰"},

                {key:36,name:"漿果叢", des:"結滿了果實"},

                {key:37,name:"採摘的花", des:"小花"},

                {key:38,name:"採摘的草", des:"小草"},

                {key:39,name:"漿果", des:"果實好好吃"},

                {key:40,name:"挖的漿果叢(拔)", des:"被挖起來的漿果叢"},

                {key:41,name:"挖的花(拔)", des:"被拔起來的花"},

                {key:42,name:"挖的樹", des:"被拔起來的樹"},

                {key:43,name:"挖的草(拔)", des:"被拔起來的草"},

                {key:44,name:"挖的漿果叢(未拔)", des:"!!!!"},

                {key:45,name:"挖的花(未拔)", des:"!!!!"},

                {key:46,name:"挖的草(未拔)", des:"!!!!"},

                {key:47,name:"挖的樹苗(拔)", des:"粗壯的小樹苗"},

                {key:48,name:"挖的樹苗(未拔)", des:"小樹苗好可愛"},

                {key:49,name:"蝙蝠翅膀", des:"會飛喔"},

                {key:50,name:"眼球", des:"好好吃"},

                {key:51,name:"蜂蜜", des:"好甜"},

                {key:52,name:"大肉", des:"肥肥"},

                {key:53,name:"怪物肉", des:"3分熟左右"},

                {key:54,name:"黑漿果", des:"上面長滿了黴菌，好臭，不能吃了。"},

            ];

    }

    this.title\_string = "";

    this.des\_string = "";

    this.showUpdate = function(item\_number){

        if(item\_number != null){

            if(item\_number <= 54 && item\_number >=1 && item\_number){

                this.title\_string = this.description[item\_number-1].name;

                this.des\_string = this.description[item\_number-1].des;

            }else{

                this.title\_string = "";

                this.des\_string = "";

            }

        }else{

            this.title\_string = "";

            this.des\_string = "";

        }

    }

    this.draw = function(ctx){

        if(this.title\_string != ""){

            ctx.beginPath();

            ctx.rect(15, 557, 250, 330);

            ctx.fillStyle = "brown";

            ctx.fill();

            this.back\_ground\_picture.draw(ctx);

            ctx.font = "25px Arial";

            ctx.fillStyle = "black";

            ctx.textAlign = 'center';

            ctx.fillText(this.title\_string, 139 ,605);

            ctx.fillText("---------------",139 ,640);

            ctx.textAlign = 'left';

            for(var i=0,j=0,k=0;i<this.des\_string.length;i++){

                if(i%8 == 0){

                    j +=30;

                    k=0;

                }

                ctx.fillText(this.des\_string.charAt(i),45 +k\*25,645+j);

                k++;

            }

        }else{

            ctx.beginPath();

            ctx.rect(0, 557, 282, 380);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

        }

    }

};

1. gameOver.js

var GameOver = Framework.Class(Framework.Level , {

    //初始化loadingProgress需要用到的圖片

    initializeProgressResource: function() {

        this.loading = new Framework.Sprite(define.imagePath + 'loading.jpg');

        this.loading.position = {x: Framework.Game.getCanvasWidth() / 2 , y: Framework.Game.getCanvasHeight() / 2};

    },

    load: function() {

        this.menu = new Framework.Sprite(define.imagePath + 'GameOver.png');

        this.url = define.materialPath + 'gate.png';

        this.open = new Framework.AnimationSprite({url:this.url, col:3 , row:4 , loop:false , speed:12});

        this.handle\_game\_over = new Handle\_game\_over();

        this.background = new Framework.Sprite(define.materialPath + 'DecorativeTile.png');

        this.frame = new Framework.Sprite(define.materialPath + 'Floral2.png');

        this.audio = new Framework.Audio({

            soundeffects: {

                mp3: define.musicPath + 'soundeffects.mp3',

            }

        });

    },

    initialize: function(){

        this.counter = 0;

        this.gameOverCount = 50;

        this.handle\_game\_over.init();

    },

    update:function(){

        this.counter++;

        this.handle\_game\_over.update();

    },

    click: function(e){

        this.handle\_game\_over.click(e);

    },

    draw: function(parentCtx) {

        this.handle\_game\_over.draw(parentCtx);

    },

});

1. handle\_game\_over.js

var Handle\_game\_over = function(){

    this.position = {

        x: Framework.Game.getCanvasWidth() / 2,

        y: Framework.Game.getCanvasHeight() / 2

    };

    this.gameClear = new Framework.Sprite(define.materialPath + 'CrossedSwords.png');

    this.gameClear.scale = 2;

    this.gameClear.position = this.position;

    this.menu = new Framework.Sprite(define.imagePath + 'GameOver.png');

    this.menu.scale = 2;

    this.menu.position = this.position;

    this.url = define.materialPath + 'gate.png';

    this.open = new Framework.AnimationSprite({url:this.url, col:3 , row:4 , loop:false , speed:12});

    this.open.scale = 11;

    this.open.position = this.position;

    this.showScore = false;

    this.init = function(){

        var audio = new Framework.Audio({

            open\_door: {

                mp3: define.musicPath + '開門.mp3',

            }

        });

        audio.play({name: 'open\_door', loop: false});

        this.open.start({ from: 0, to: 11, loop: false});

    }

    this.update = function(){

        this.open.update();

    }

    this.click = function(e){

        var audio = new Framework.Audio({

            victory: {

                mp3: define.musicPath + '勝利.mp3',

            }

        });

        if(!this.showScore && !this.open.\_start){

            this.showScore = true;

            audio.play({name: 'victory', loop: true});

            this.draw(Framework.Game.\_context);

        }

        if(Framework.Game.\_levels[2].level.map.score.scoreToDraw >= Framework.Game.\_levels[2].level.map.score.score){

            audio.stopAll();

            Framework.Game.goToLevel('menu');

        }

    }

    this.draw = function(ctx){

        if(this.open.\_start)

            this.open.draw(ctx);

        else{

            if(Framework.Game.\_levels[2].level.map.player1.character\_descruption\_point[0] <= 0)

                this.menu.draw(ctx);

            else{

                this.gameClear.draw(ctx);

                ctx.font = '90pt Algerian';

                ctx.globalAlpha=1;

                ctx.fillStyle = 'aqua';

                ctx.textBaseline = 'top';

                ctx.textAlign = 'center';

                ctx.strokeStyle = 'orange';

                ctx.lineWidth = 2.5;

                ctx.fillText("Game Clear !!!", this.position.x, this.position.y);

                ctx.strokeText("Game Clear !!!", this.position.x, this.position.y);

            }

        }

        if(this.showScore)

            Framework.Game.\_levels[2].level.map.score.drawScore();

    }

}

1. handle\_initial\_character.js

var Handle\_initial\_character = function(){

    this.character\_description = new Character\_description();

    this.alphabet = new Framework.Sprite(define.materialPath + 'alphabet.png');

    this.alphabet.scale = 0.3;

    this.alphabet.position = {x:1274, y:330};

    this.big\_white = new Framework.Sprite(define.materialPath + 'big\_white.png');

    this.big\_white.scale = 0.3;

    this.big\_white.position = {x:1274, y:330};

    this.is\_typepanel\_open = false;

    this.is\_initial = false;

    this.name = "請輸入名字";

    this.init = function(){

        this.character\_description.is\_character\_description\_open = true;

        this.character\_description.experience = 0;

        this.character\_description.capabilityt\_point = 5;

        for(var i = 0;i < this.character\_description.character\_descruption\_point.length;i++){

            this.character\_description.character\_descruption\_point[i] = 5;

        }

    }

    this.update = function(){

        if(this.character\_description.capabilityt\_point == 0)

            this.is\_initial = true;

    }

    this.is\_alphabet\_touch = false;

    this.alphabet\_touch\_anchor = {x:0, y:0};

    this.draw = function(ctx){

        this.character\_description.draw(ctx);

        ctx.beginPath();

        ctx.rect(818, 220, 250, 50);

        ctx.fillStyle = "#00FFFF";

        ctx.fill();

        ctx.font = "25px Arial";

        ctx.fillStyle = "black";

        ctx.textAlign = 'center';

        ctx.fillText(this.name, 818+125 ,255);

        if(this.is\_typepanel\_open){

            this.big\_white.draw(ctx);

            if(this.is\_alphabet\_touch){

                ctx.beginPath();

                ctx.rect(this.alphabet\_touch\_anchor.x, this.alphabet\_touch\_anchor.y, 55, 35);

                ctx.fillStyle = "silver";

                ctx.fill();

            }

            this.alphabet.draw(ctx);

        }

        else{

            ctx.beginPath();

            ctx.rect(1207, 220, 300, 500);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

        }

    }

    this.mousemove = function(e){

        if(this.is\_typepanel\_open){

            this.is\_alphabet\_touch = true;

            if(e.x>= 1086 && e.x<= 1145){

                if(e.y>= 226 && e.y<= 262)

                    this.alphabet\_touch\_anchor = {x:1086, y:226};

                else if(e.y>= 269 && e.y<= 307)

                    this.alphabet\_touch\_anchor = {x:1086, y:269};

                else if(e.y>= 311 && e.y<= 349)

                    this.alphabet\_touch\_anchor = {x:1086, y:311};

                else if(e.y>= 353 && e.y<= 391)

                    this.alphabet\_touch\_anchor = {x:1086, y:353};

                else if(e.y>= 396 && e.y<= 433)

                    this.alphabet\_touch\_anchor = {x:1086, y:396};

                else

                    this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1149 && e.x<= 1208){

                if(e.y>= 226 && e.y<= 262)

                    this.alphabet\_touch\_anchor = {x:1149, y:226};

                else if(e.y>= 269 && e.y<= 307)

                    this.alphabet\_touch\_anchor = {x:1149, y:269};

                else if(e.y>= 311 && e.y<= 349)

                    this.alphabet\_touch\_anchor = {x:1149, y:311};

                else if(e.y>= 353 && e.y<= 391)

                    this.alphabet\_touch\_anchor = {x:1149, y:353};

                else if(e.y>= 396 && e.y<= 433)

                    this.alphabet\_touch\_anchor = {x:1149, y:396};

                else

                    this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1214 && e.x<= 1271){

                if(e.y>= 226 && e.y<= 262)

                    this.alphabet\_touch\_anchor = {x:1214, y:226};

                else if(e.y>= 269 && e.y<= 307)

                    this.alphabet\_touch\_anchor = {x:1214, y:269};

                else if(e.y>= 311 && e.y<= 349)

                    this.alphabet\_touch\_anchor = {x:1214, y:311};

                else if(e.y>= 353 && e.y<= 391)

                    this.alphabet\_touch\_anchor = {x:1214, y:353};

                else if(e.y>= 396 && e.y<= 433)

                    this.alphabet\_touch\_anchor = {x:1214, y:396};

                else

                    this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1276 && e.x<= 1336){

                if(e.y>= 226 && e.y<= 262)

                    this.alphabet\_touch\_anchor = {x:1276, y:226};

                else if(e.y>= 269 && e.y<= 307)

                    this.alphabet\_touch\_anchor = {x:1276, y:269};

                else if(e.y>= 311 && e.y<= 349)

                    this.alphabet\_touch\_anchor = {x:1276, y:311};

                else if(e.y>= 353 && e.y<= 391)

                    this.alphabet\_touch\_anchor = {x:1276, y:353};

                else if(e.y>= 396 && e.y<= 433)

                    this.alphabet\_touch\_anchor = {x:1276, y:396};

                else

                    this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1339 && e.x<= 1398){

                if(e.y>= 226 && e.y<= 262)

                    this.alphabet\_touch\_anchor = {x:1339, y:226};

                else if(e.y>= 269 && e.y<= 307)

                    this.alphabet\_touch\_anchor = {x:1339, y:269};

                else if(e.y>= 311 && e.y<= 349)

                    this.alphabet\_touch\_anchor = {x:1339, y:311};

                else if(e.y>= 353 && e.y<= 391)

                    this.alphabet\_touch\_anchor = {x:1339, y:353};

                else if(e.y>= 396 && e.y<= 433)

                    this.alphabet\_touch\_anchor = {x:1339, y:396};

                else

                    this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1403 && e.x<= 1461){

                if(e.y>= 226 && e.y<= 262)

                    this.alphabet\_touch\_anchor = {x:1403, y:226};

                else if(e.y>= 269 && e.y<= 307)

                    this.alphabet\_touch\_anchor = {x:1403, y:269};

                else

                    this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }else{

                this.is\_alphabet\_touch = false;

                this.draw(Framework.Game.\_context);

            }

        }

    }

    this.click = function(e){

        var audio = new Framework.Audio({

            start\_game: {

                mp3: define.musicPath + '遊戲開場音樂.mp3',

            }, keyup:{

                mp3: define.musicPath + '按按鍵.mp3',

            }, song2:{

                mp3: define.musicPath + '刀劍神域op1《crossing field》鋼琴曲.mp3',

            }

        });

        audio.play({name: 'keyup', loop: false});

        if(this.is\_typepanel\_open){

            this.is\_alphabet\_touch = true;

            if(e.x>= 1086 && e.x<= 1145){

                if(e.y>= 226 && e.y<= 262)

                    this.name += "A";

                else if(e.y>= 269 && e.y<= 307)

                    this.name += "B";

                else if(e.y>= 311 && e.y<= 349)

                    this.name += "C";

                else if(e.y>= 353 && e.y<= 391)

                    this.name += "D";

                else if(e.y>= 396 && e.y<= 433)

                    this.name += "E";

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1149 && e.x<= 1208){

                if(e.y>= 226 && e.y<= 262)

                    this.name += "F";

                else if(e.y>= 269 && e.y<= 307)

                    this.name += "G";

                else if(e.y>= 311 && e.y<= 349)

                    this.name += "H";

                else if(e.y>= 353 && e.y<= 391)

                    this.name += "I";

                else if(e.y>= 396 && e.y<= 433)

                    this.name += "J";

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1214 && e.x<= 1271){

                if(e.y>= 226 && e.y<= 262)

                    this.name += "K";

                else if(e.y>= 269 && e.y<= 307)

                    this.name += "L";

                else if(e.y>= 311 && e.y<= 349)

                    this.name += "M";

                else if(e.y>= 353 && e.y<= 391)

                    this.name += "N";

                else if(e.y>= 396 && e.y<= 433)

                    this.name += "O";

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1276 && e.x<= 1336){

                if(e.y>= 226 && e.y<= 262)

                    this.name += "P";

                else if(e.y>= 269 && e.y<= 307)

                    this.name += "Q";

                else if(e.y>= 311 && e.y<= 349)

                    this.name += "R";

                else if(e.y>= 353 && e.y<= 391)

                    this.name += "S";

                else if(e.y>= 396 && e.y<= 433)

                    this.name += "T";

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1339 && e.x<= 1398){

                if(e.y>= 226 && e.y<= 262)

                    this.name += "U";

                else if(e.y>= 269 && e.y<= 307)

                    this.name += "V";

                else if(e.y>= 311 && e.y<= 349)

                    this.name += "W";

                else if(e.y>= 353 && e.y<= 391)

                    this.name += "X";

                else if(e.y>= 396 && e.y<= 433)

                    this.name += "Y";

                this.draw(Framework.Game.\_context);

            }else if(e.x>= 1403 && e.x<= 1461){

                if(e.y>= 226 && e.y<= 262)

                    this.name += "Z";

                else if(e.y>= 269 && e.y<= 307)

                    this.is\_typepanel\_open = false;

                this.draw(Framework.Game.\_context);

            }

        }

        if(e.x>= 818 && e.x<= 997 && e.y >= 221 && e.y<= 270){

            this.name = ""

            this.draw(Framework.Game.\_context);

            this.is\_typepanel\_open = true;

        }

        if(e.x >= 470 && e.x <=510){

            if(e.y >=530 && e.y<= 570)

                this.character\_description.isChangeCapability(2);

            else if(e.y >=612 && e.y<= 652)

                this.character\_description.isChangeCapability(3);

            else if(e.y >=690 && e.y<= 730)

                this.character\_description.isChangeCapability(4);

        }else if(e.x >= 822 && e.x <=861){

            if(e.y >=382 && e.y<= 425)

                this.character\_description.isChangeCapability(5);

            else if(e.y >=461 && e.y<= 502)

                this.character\_description.isChangeCapability(6);

            else if(e.y >=622 && e.y<= 661)

                this.character\_description.isChangeCapability(8);

        }

    }

}

1. item\_armor.js

var Item\_armor = function() {

    this.item\_armor = new Framework.Sprite(define.materialPath + 'item\_armor.png');

    this.item\_armor.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.durability = 100;

    this.item\_num = 23;

    this.item\_can\_be\_picked = true;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "body";

    this.deffense\_point = 40;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_armor.draw(ctx);

    }

    this.reduceDurability = function(){

        this.durability -= 2;

    }

};

Object.defineProperty(Item\_armor.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_armor.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_bee\_sting.js

var Item\_bee\_sting = function() {

    this.item\_bee\_sting = new Framework.Sprite(define.materialPath + 'item\_bee\_sting.png');

    this.item\_bee\_sting.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 11;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_bee\_sting.draw(ctx);

    }

};

Object.defineProperty(Item\_bee\_sting.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_bee\_sting.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_berry.js

var Item\_berry = function() {

    this.item\_berry = new Framework.Sprite(define.materialPath + 'item\_berry.png');

    this.item\_berry.scale = 2;

    this.item\_num = 39;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_berry, 3);

};

Item\_berry.prototype = new Food\_base();

1. item\_black\_berry.js

var Item\_black\_berry = function() {

    this.item\_black\_berry = new Framework.Sprite(define.materialPath + 'item\_black\_berry.png');

    this.item\_black\_berry.scale = 2;

    this.item\_num = 54;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_black\_berry, 3);

};

Item\_black\_berry.prototype = new Food\_base();

1. item\_blank.js

var Item\_blank = function() {

    this.item\_blank = new Framework.Sprite(define.materialPath + 'item\_blank.png');

    this.mapPosition = {x:0, y:0};

    this.item\_can\_be\_picked = false;

    this.item\_num = 0;

    this.isRegenerate = false;

    this.draw = function(ctx){

    }

};

Object.defineProperty(Item\_blank.prototype, 'position', {

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_blank.position = {x: 0, y: 0};

    }

});

1. item\_branch.js

var Item\_branch = function() {

    this.item\_branch = new Framework.Sprite(define.materialPath + 'item\_branch.png');

    this.item\_branch.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_can\_be\_picked = true;

    this.item\_num = 4;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.draw = function(ctx){

        this.item\_branch.draw(ctx);

    }

};

Object.defineProperty(Item\_branch.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_branch.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_bush.js

var Item\_bush = function() {

    this.item\_bush = new Framework.Sprite(define.materialPath + 'item\_bush.png');

    this.item\_bush\_picked = new Framework.Sprite(define.materialPath + 'item\_bush\_picked.png');

    this.item\_bush.scale = 2;

    this.item\_bush\_picked.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = true;

    this.item\_num = 36;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = false;

    this.regeneration\_time = 3000;

    this.reset = function(){

        setTimeout(()=>{  this.status = true }, this.regeneration\_time);

    }

    this.update = function(){

        this.status = false;

        this.reset();

    }

    this.draw = function(ctx){

        if(this.status)

            this.item\_bush.draw(ctx);

        else

            this.item\_bush\_picked.draw(ctx);

    }

};

Object.defineProperty(Item\_bush.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_bush.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

        this.item\_bush\_picked.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_bush\_dig.js

var Item\_bush\_dig = function() {

    this.item\_bush\_dig = new Framework.Sprite(define.materialPath + 'item\_bush\_dig.png');

    this.item\_bush\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 40;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_bush\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_bush\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_bush\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_bush\_growed\_dig.js

var Item\_bush\_growed\_dig = function() {

    this.item\_bush\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_bush\_growed\_dig.png');

    this.item\_bush\_growed\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 44;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_bush\_growed\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_bush\_growed\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_bush\_growed\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_camp.js

var Item\_camp = function() {

    this.item\_camp = new Framework.Sprite(define.materialPath + 'item\_camp.png');

    this.item\_camp.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 33;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "tool";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_camp.draw(ctx);

    }

};

Object.defineProperty(Item\_camp.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_camp.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_campfire.js

var Item\_campfire = function() {

    this.item\_campfire = new Framework.Sprite(define.materialPath + 'item\_campfire.png');

    this.item\_campfire.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 34;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "tool";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_campfire.draw(ctx);

    }

};

Object.defineProperty(Item\_campfire.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_campfire.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_droplet.js

var Item\_droplet = function() {

    this.item\_droplet = new Framework.Sprite(define.materialPath + 'item\_droplet.png');

    this.item\_droplet.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 14;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_droplet.draw(ctx);

    }

};

Object.defineProperty(Item\_droplet.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_droplet.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_eyeball.js

var Item\_eyeball = function() {

    this.item\_eyeball = new Framework.Sprite(define.materialPath + 'Eyeball.png');

    this.item\_eyeball.scale = 0.8;

    this.item\_num = 50;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_eyeball, 1);

};

Item\_eyeball.prototype = new Food\_base();

1. item\_fake.js

var Item\_fake = function() {

    this.mapPosition = {x:0, y:0};

    this.item\_can\_be\_picked = false;

    this.item\_num = -3;

    this.isRegenerate = false;

    this.draw = function(ctx){

    }

};

Object.defineProperty(Item\_fake.prototype, 'position', {

    set: function(newValue) {

        this.mapPosition = newValue;

    }

});

1. item\_firebundle.js

var Item\_firebundle = function() {

    this.item\_firebundle = new Framework.Sprite(define.materialPath + 'item\_firebundle.png');

    this.item\_firebundle\_no = new Framework.Sprite(define.materialPath + 'item\_firebundle\_no.png');

    this.item\_firebundle.scale = 1.5;

    this.item\_firebundle\_no.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 32;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "hand";

    this.inEquipmentbar = false;

    this.item\_can\_be\_picked = true;

    this.amount = 1;

    this.attack\_point = 1;

    this.magic\_attack\_point = 0;

    this.arror\_attack\_point = 0;

    this.reduceDurability = function(){

        if(this.inEquipmentbar){

            this.durability -= 1;

            setTimeout(()=>{

                this.reduceDurability();

            },1000);

        }

    }

    this.draw = function(ctx){

        if(this.inEquipmentbar)

            this.item\_firebundle.draw(ctx);

        else

            this.item\_firebundle\_no.draw(ctx);

    }

};

Object.defineProperty(Item\_firebundle.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_firebundle.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

        this.item\_firebundle\_no.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_fish.js

var Item\_fish = function() {

    this.item\_fish = new Framework.Sprite(define.materialPath + 'item\_fish.png');

    this.item\_fish.scale = 0.8;

    this.item\_num = 49;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_fish, 1);

};

Item\_fish.prototype = new Food\_base();

1. item\_flint.js

var Item\_flint = function() {

    this.item\_flint = new Framework.Sprite(define.materialPath + 'item\_flint.png');

    this.item\_flint.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 8;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_flint.draw(ctx);

    }

};

Object.defineProperty(Item\_flint.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_flint.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_flower.js

var Item\_flower = function() {

    this.item\_flower = new Framework.Sprite(define.materialPath + 'item\_flower.png');

    this.item\_flower\_pulled = new Framework.Sprite(define.materialPath + 'item\_flower\_pulled.png');

    this.item\_flower.scale = 2;

    this.item\_flower\_pulled.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = true;

    this.item\_num = 1;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.regeneration\_time = 3000;

    this.update = function(){

        this.status = false;

        setTimeout(()=>{  this.status = true}, this.regeneration\_time);

    }

    this.draw = function(ctx){

        if(this.status)

            this.item\_flower.draw(ctx);

        else

            this.item\_flower\_pulled.draw(ctx);

    }

};

Object.defineProperty(Item\_flower.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_flower.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

        this.item\_flower\_pulled.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_flower\_dig.js

var Item\_flower\_dig = function() {

    this.item\_flower\_dig = new Framework.Sprite(define.materialPath + 'item\_flower\_dig.png');

    this.item\_flower\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 41;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_flower\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_flower\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_flower\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_flower\_growed\_dig.js

var Item\_flower\_growed\_dig = function() {

    this.item\_flower\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_flower\_growed\_dig.png');

    this.item\_flower\_growed\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 45;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

    }

    this.draw = function(ctx){

        this.item\_flower\_growed\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_flower\_growed\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_flower\_growed\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_flower\_picked.js

var Item\_flower\_picked = function() {

    this.item\_flower\_picked = new Framework.Sprite(define.materialPath + 'item\_flower\_picked.png');

    this.item\_flower\_picked.scale = 2;

    this.item\_num = 37;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_flower\_picked, 1);

};

Item\_flower\_picked.prototype = new Food\_base();

1. item\_gold.js

var Item\_gold = function() {

    this.item\_gold = new Framework.Sprite(define.materialPath + 'item\_gold.png');

    this.item\_gold.scale = 0.7;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 9;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_gold.draw(ctx);

    }

};

Object.defineProperty(Item\_gold.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_gold.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_grass.js

var Item\_grass = function() {

    this.item\_grass = new Framework.Sprite(define.materialPath + 'item\_grass.png');

    this.item\_grass\_mowed = new Framework.Sprite(define.materialPath + 'item\_grass\_mowed.png');

    this.item\_grass.scale = 2;

    this.item\_grass\_mowed.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = true;

    this.item\_num = 6;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.regeneration\_time = 3000;

    this.reset = function(){

        setTimeout(()=>{  this.status = true}, this.regeneration\_time);

    }

    this.update = function(){

        this.status = false;

        this.reset();

    }

    this.draw = function(ctx){

        if(this.status){

            this.item\_grass.draw(ctx);

        }else{

            this.item\_grass\_mowed.draw(ctx);

        }

    }

};

Object.defineProperty(Item\_grass.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_grass.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

        this.item\_grass\_mowed.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_grass\_dig.js

var Item\_grass\_dig = function() {

    this.item\_grass\_dig = new Framework.Sprite(define.materialPath + 'item\_grass\_dig.png');

    this.item\_grass\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 43;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

    }

    this.draw = function(ctx){

        this.item\_grass\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_grass\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_grass\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_grass\_growed\_dig.js

var Item\_grass\_growed\_dig = function() {

    this.item\_grass\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_grass\_growed\_dig.png');

    this.item\_grass\_growed\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 46;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

    }

    this.draw = function(ctx){

        this.item\_grass\_growed\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_grass\_growed\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_grass\_growed\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_grass\_picked.js

var Item\_grass\_picked = function() {

    this.item\_grass\_picked = new Framework.Sprite(define.materialPath + 'item\_grass\_picked.png');

    this.item\_grass\_picked.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 38;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_grass\_picked.draw(ctx);

    }

};

Object.defineProperty(Item\_grass\_picked.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_grass\_picked.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_helmat.js

var Item\_helmat = function() {

    this.item\_helmat = new Framework.Sprite(define.materialPath + 'item\_helmat.png');

    this.item\_helmat.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 22;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "head";

    this.deffense\_point = 40;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_helmat.draw(ctx);

    }

    this.reduceDurability = function(){

        this.durability -= 2;

    }

};

Object.defineProperty(Item\_helmat.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_helmat.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_honey.js

var Item\_honey = function() {

    this.item\_honey = new Framework.Sprite(define.materialPath + 'Honey.png');

    this.item\_honey.scale = 0.8;

    this.item\_num = 51;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_honey, 2);

};

Item\_honey.prototype = new Food\_base();

1. item\_ice.js

var Item\_ice = function() {

    this.item\_ice = new Framework.Sprite(define.materialPath + 'item\_ice.png');

    this.item\_ice.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 35;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_ice.draw(ctx);

    }

};

Object.defineProperty(Item\_ice.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_ice.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_king\_wand.js

var Item\_king\_wand = function() {

    this.item\_king\_wand = new Framework.Sprite(define.materialPath + 'item\_king\_wand.png');

    this.item\_king\_wand.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 27;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "hand";

    this.item\_can\_be\_picked = true;

    this.attack\_point = 0;

    this.magic\_attack\_point = 20;

    this.arror\_attack\_point = 0

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_king\_wand.draw(ctx);

    }

    this.reduceDurability = function(){

        this.durability -= 20;

    }

};

Object.defineProperty(Item\_king\_wand.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_king\_wand.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_lamp.js

var Item\_lamp = function() {

    this.item\_lamp = new Framework.Sprite(define.materialPath + 'item\_lamp.png');

    this.item\_lamp.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 31;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "hand";

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_lamp.draw(ctx);

    }

};

Object.defineProperty(Item\_lamp.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_lamp.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_meat.js

var Item\_meat = function() {

    this.item\_meat = new Framework.Sprite(define.materialPath + 'Meat.png');

    this.item\_meat.scale = 0.8;

    this.item\_num = 52;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_meat, 1);

};

Item\_meat.prototype = new Food\_base();

1. item\_monster\_meat.js

var Item\_monster\_meat = function() {

    this.item\_monster\_meat = new Framework.Sprite(define.materialPath + 'Monster\_Meat.png');

    this.item\_monster\_meat.scale = 0.8;

    this.item\_num = 53;

    this.hungerAddition = 20;

    this.healthAddition = 20;

    //圖片，數量

    this.init(this.item\_monster\_meat, 1);

};

Item\_monster\_meat.prototype = new Food\_base();

1. item\_pigskin.js

var Item\_pigskin = function() {

    this.item\_pigskin = new Framework.Sprite(define.materialPath + 'item\_pigskin.png');

    this.item\_pigskin.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 10;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_pigskin.draw(ctx);

    }

};

Object.defineProperty(Item\_pigskin.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_pigskin.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_rope.js

var Item\_rope = function() {

    this.item\_rope = new Framework.Sprite(define.materialPath + 'item\_rope.png');

    this.item\_rope.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 13;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_rope.draw(ctx);

    }

};

Object.defineProperty(Item\_rope.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_rope.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_sapling.js

var Item\_sapling = function() {

    this.item\_sapling = new Framework.Sprite(define.materialPath + 'item\_sapling.png');

    this.item\_sapling\_pulled = new Framework.Sprite(define.materialPath + 'item\_sapling\_pulled.png');

    this.item\_sapling.scale = 0.6;

    this.item\_sapling\_pulled.scale = 0.4;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = true;

    this.item\_num = -4;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.regeneration\_time = 3000;

    this.update = function(){

        this.status = false;

        setTimeout(()=>{  this.status = true}, this.regeneration\_time);

    }

    this.draw = function(ctx){

        if(this.status)

            this.item\_sapling.draw(ctx);

        else

            this.item\_sapling\_pulled.draw(ctx);

    }

};

Object.defineProperty(Item\_sapling.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_sapling.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64 - 15};

        this.item\_sapling\_pulled.position = {x: this.mapPosition.x \* 64 -20, y: this.mapPosition.y \* 64 + 5};

    }

});

1. item\_sapling\_dig.js

var Item\_sapling\_dig = function() {

    this.item\_sapling\_dig = new Framework.Sprite(define.materialPath + 'item\_sapling\_dig.png');

    this.item\_sapling\_dig.scale = 0.4;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 47;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_sapling\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_sapling\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_sapling\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_sapling\_growed\_dig.js

var Item\_sapling\_growed\_dig = function() {

    this.item\_sapling\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_sapling\_growed\_dig.png');

    this.item\_sapling\_growed\_dig.scale = 0.45;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 48;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_sapling\_growed\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_sapling\_growed\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_sapling\_growed\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_snow\_ball.js

var Item\_snow\_ball = function() {

    this.item\_snow\_ball = new Framework.Sprite(define.materialPath + 'item\_snow\_ball.png');

    this.item\_snow\_ball.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 12;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_snow\_ball.draw(ctx);

    }

};

Object.defineProperty(Item\_snow\_ball.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_snow\_ball.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_space\_wand.js

var Item\_space\_wand = function() {

    this.item\_space\_wand = new Framework.Sprite(define.materialPath + 'item\_space\_wand.png');

    this.item\_space\_wand.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 28;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "hand";

    this.attack\_point = 0;

    this.magic\_attack\_point = 60;

    this.arror\_attack\_point = 0;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_space\_wand.draw(ctx);

    }

    this.reduceDurability = function(){

        this.durability -= 100;

    }

};

Object.defineProperty(Item\_space\_wand.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_space\_wand.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_spider\_web.js

var Item\_spider\_web = function() {

    this.item\_spider\_web = new Framework.Sprite(define.materialPath + 'item\_spider\_web.png');

    this.item\_spider\_web.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 2;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        if(this.status)

            this.item\_spider\_web.draw(ctx);

    }

};

Object.defineProperty(Item\_spider\_web.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_spider\_web.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_stone.js

var Item\_stone = function() {

    this.item\_stone = new Framework.Sprite(define.materialPath + 'item\_stone.png');

    this.item\_stone\_dig = new Framework.Sprite(define.materialPath + 'item\_stone\_dig.png');

    this.item\_stone.scale = 2;

    this.item\_stone\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = false;

    this.isRegenerate = false;

    this.item\_num = 3;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.count = 0;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.count += 1;

    }

    this.draw = function(ctx){

        if(this.count < 3)

            this.item\_stone.draw(ctx);

        else

            this.item\_stone\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_stone.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_stone.position = {x: this.mapPosition.x \* 64 + 30, y: this.mapPosition.y \* 64};

        this.item\_stone\_dig.position = {x: this.mapPosition.x \* 64 + 30, y: this.mapPosition.y \* 64};

    }

});

1. item\_tree\_dig.js

var Item\_tree\_dig = function() {

    this.item\_tree\_dig = new Framework.Sprite(define.materialPath + 'item\_tree\_dig.png');

    this.item\_tree\_dig.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 42;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "plant";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.item\_tree\_dig.draw(ctx);

    }

};

Object.defineProperty(Item\_tree\_dig.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_tree\_dig.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_waikei\_homework.js

var Item\_waikei\_homework = function() {

    this.item\_waikei\_homework = new Framework.Sprite(define.materialPath + 'item\_waikei\_homework.png');

    this.item\_waikei\_homework.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 5;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_waikei\_homework.draw(ctx);

    }

};

Object.defineProperty(Item\_waikei\_homework.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_waikei\_homework.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_wood.js

var Item\_wood = function() {

    this.item\_wood = new Framework.Sprite(define.materialPath + 'item\_wood.png');

    this.item\_wood.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 7;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_wood.draw(ctx);

    }

};

Object.defineProperty(Item\_wood.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_wood.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. item\_wood\_armor.js

var Item\_wood\_armor = function() {

    this.item\_wood\_armor = new Framework.Sprite(define.materialPath + 'item\_wood\_armor.png');

    this.item\_wood\_armor.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = false;

    this.item\_num = 24;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "equipment";

    //可疊加物件有amount 不可疊加有durability

    this.durability = 100;

    this.place = "body";

    this.deffense\_point = 60;

    this.item\_can\_be\_picked = true;

    this.update = function(){

        this.status = false;

    }

    this.draw = function(ctx){

        this.item\_wood\_armor.draw(ctx);

    }

    this.reduceDurability = function(){

        this.durability -= 2;

    }

};

Object.defineProperty(Item\_wood\_armor.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.item\_wood\_armor.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. level\_up\_animation.js

var Level\_up\_animation = function() {

    this.url = define.materialPath + 'levelup.png';

    this.level\_up\_animation = new Framework.AnimationSprite({url:this.url, col:5 , row:2 , loop:false , speed:5});

    this.level\_up\_animation.scale = 1;

    this.level\_up\_animation.position = {x:12\*64, y:5.5\*64};

    this.update = function(){

        this.level\_up\_animation.update();

    }

    this.draw = function(ctx){

        this.level\_up\_animation.draw(ctx);

    }

    this.start = function(){

        this.level\_up\_animation.start({ from: 0, to: 9, loop: false});

    }

};

1. level2\_change.js

var Level2\_change = Framework.Class(Framework.Level , {

    initializeProgressResource: function() {

        this.loading = new Framework.Sprite(define.imagePath + 'loading.jpg');

        this.loading.position = {x: Framework.Game.getCanvasWidth() / 2 , y: Framework.Game.getCanvasHeight() / 2};

    },

    loadingProgress: function(ctx, requestInfo) {

        this.loading.draw(ctx);

        ctx.font ='90px Arial';

        ctx.textAlign = 'center';

        ctx.fillStyle = 'white';

        ctx.fillText(Math.round(requestInfo.percent) + '%' , ctx.canvas.width / 2 , ctx.canvas.height / 2 + 300);

    },

    load: function() {

        this.map = new World\_map();

        this.map.load();

        this.map.addMonsterRandom(10);

    },

    initialize: function() {

        this.map.init();

        this.map.setPlayerPosition({x:13,y:7});

    },

    update: function() {

        this.map.update();

    },

    draw:function(parentCtx){

        this.map.draw(parentCtx);

    },

    keydown:function(e, list){

        Framework.DebugInfo.Log.warning(e.key);

        this.map.keydown(e, list);

        if(e.key === 'F11') {

            if(!this.isFullScreen) {

                Framework.Game.fullScreen();

                this.isFullScreen = true;

            } else {

                Framework.Game.exitFullScreen();

                this.isFullScreen = false;

            }

        }

    },

    keyup:function(e, list){

        this.map.keyup(e, list);

    },

    click: function (e) {

        this.map.click(e);

    },

    mousemove: function(e) {

        this.map.mousemove(e);

    },

});

1. loadGame.js

//立即執行函式, 並封裝所有變數避免衝突

var loadGameEnd;

(function(){

    //動態依序載入JS

    //ref: http://blog.darkthread.net/blogs/darkthreadtw/archive/2009/01/15/4061.aspx

    var  importJS = function(jsConf, src, lookFor) {

        var headID = document.getElementsByTagName("head")[0];

        var newJs = document.createElement('script');

        newJs.type = 'text/javascript';

        newJs.src= jsConf[0].src;

        headID.appendChild(newJs);

        wait\_for\_script\_load(jsConf, function() {

            jsConf.splice(0, 1);

            if(jsConf.length > 0) {

                importJS(jsConf, lookFor);

                if(typeof blanket != "undefined"){

                    blanket.utils.cache[jsConf[0].src] = {};

                    blanket.utils.attachScript({url: jsConf[0].src}, function (content) {

                        blanket.instrument({inputFile: content, inputFileName: jsConf[0].src},function (instrumented) {

                            blanket.utils.cache[jsConf[0].src].loaded = true;

                            blanket.utils.blanketEval(instrumented);

                            blanket.requiringFile(jsConf[0].src, true);

                        });

                    });

                }

            }else

            {

                loadGameEnd = true;

            }

        });

    }

    var wait\_for\_script\_load = function(jsConf, callback) {

        var interval = setInterval(function() {

            if (typeof jsConf[0].lookFor === 'undefined') {

                jsConf[0].lookFor = '';

            }

            if (jsConf[0].lookFor === '') {

                clearInterval(interval);

                callback();

            } else if (eval("typeof " + jsConf[0].lookFor) !== 'undefined') {

                    clearInterval(interval);

                    callback();

                }

            }, 50);

    }

    //陣列和載入JS檔的順序相同, lookFor為在要載入的檔案中,

    //有用到的全域變數, importJS這個function, 會在找到lookFor的變數後

    //才會繼續loading下一個檔案, 如果沒有需要lookFor, 則以空字串代表

    var listScript =

    [

        { src: 'game\_sample/js/define.js', lookFor: 'define' },

        { src: 'game\_sample/js/myMenu.js', lookFor: 'MyMenu' },

        { src: 'game\_sample/js/myMenu\_tutorial.js', lookFor: 'MyMenu\_tutorial' },

        { src: 'game\_sample/js/myMenu\_gameteam.js', lookFor: 'MyMenu\_gameteam' },

        { src: 'game\_sample/js/constants.js', lookFor: 'Constants' },

        { src: 'game\_sample/js/score.js', lookFor: 'Score' },

        { src: 'game\_sample/js/bombMan.js', lookFor: 'BombMan' },

        { src: 'game\_sample/js/food\_base.js', lookFor: 'Food\_base' },

        { src: 'game\_sample/js/item\_lamp.js', lookFor: 'Item\_lamp' },

        { src: 'game\_sample/js/map\_item\_tree.js', lookFor: 'Map\_item\_tree' },

        { src: 'game\_sample/js/item\_grass.js', lookFor: 'Item\_grass' },

        { src: 'game\_sample/js/item\_grass\_picked.js', lookFor: 'Item\_grass\_picked' },

        { src: 'game\_sample/js/item\_rope.js', lookFor: 'Item\_rope' },

        { src: 'game\_sample/js/item\_stone.js', lookFor: 'Item\_stone' },

        { src: 'game\_sample/js/item\_droplet.js', lookFor: 'Item\_droplet' },

        { src: 'game\_sample/js/item\_bee\_sting.js', lookFor: 'Item\_bee\_sting' },

        { src: 'game\_sample/js/item\_gold.js', lookFor: 'Item\_gold' },

        { src: 'game\_sample/js/item\_bush.js', lookFor: 'Item\_bush' },

        { src: 'game\_sample/js/item\_ice.js', lookFor: 'Item\_ice' },

        { src: 'game\_sample/js/item\_armor.js', lookFor: 'Item\_armor' },

        { src: 'game\_sample/js/item\_wood\_armor.js', lookFor: 'Item\_wood\_armor' },

        { src: 'game\_sample/js/item\_helmat.js', lookFor: 'Item\_helmat' },

        { src: 'game\_sample/js/item\_flint.js', lookFor: 'Item\_flint' },

        { src: 'game\_sample/js/item\_pigskin.js', lookFor: 'Item\_pigskin' },

        { src: 'game\_sample/js/item\_snow\_ball.js', lookFor: 'Item\_snow\_ball' },

        { src: 'game\_sample/js/item\_space\_wand.js', lookFor: 'Item\_space\_wand' },

        { src: 'game\_sample/js/item\_king\_wand.js', lookFor: 'Item\_king\_wand' },

        { src: 'game\_sample/js/item\_wood.js', lookFor: 'Item\_wood' },

        { src: 'game\_sample/js/item\_campfire.js', lookFor: 'Item\_campfire' },

        { src: 'game\_sample/js/item\_camp.js', lookFor: 'Item\_camp' },

        { src: 'game\_sample/js/item\_firebundle.js', lookFor: 'Item\_firebundle' },

        { src: 'game\_sample/js/item\_flower.js', lookFor: 'Item\_flower' },

        { src: 'game\_sample/js/item\_flower\_picked.js', lookFor: 'Item\_flower\_picked' },

        { src: 'game\_sample/js/item\_berry.js', lookFor: 'Item\_berry' },

        { src: 'game\_sample/js/item\_black\_berry.js', lookFor: 'Item\_black\_berry' },

        { src: 'game\_sample/js/item\_waikei\_homework.js', lookFor: 'Item\_waikei\_homework' },

        { src: 'game\_sample/js/item\_spider\_web.js', lookFor: 'Item\_spider\_web' },

        { src: 'game\_sample/js/item\_flower\_growed\_dig.js', lookFor: 'Item\_flower\_growed\_dig' },

        { src: 'game\_sample/js/item\_honey.js', lookFor: 'Item\_honey' },

        { src: 'game\_sample/js/item\_meat.js', lookFor: 'Item\_meat' },

        { src: 'game\_sample/js/item\_monster\_meat.js', lookFor: 'Item\_monster\_meat' },

        { src: 'game\_sample/js/item\_flower\_growed\_dig.js', lookFor: 'Item\_flower\_growed\_dig' },

        { src: 'game\_sample/js/item\_sapling.js', lookFor: 'Item\_sapling' },

        { src: 'game\_sample/js/item\_grass\_growed\_dig.js', lookFor: 'Item\_grass\_growed\_dig' },

        { src: 'game\_sample/js/item\_bush\_growed\_dig.js', lookFor: 'Item\_bush\_growed\_dig' },

        { src: 'game\_sample/js/item\_sapling\_growed\_dig.js', lookFor: 'Item\_sapling\_growed\_dig' },

        { src: 'game\_sample/js/item\_sapling\_dig.js', lookFor: 'Item\_sapling\_dig' },

        { src: 'game\_sample/js/drama0.js', lookFor: 'Drama0' },

        { src: 'game\_sample/js/item\_blank.js', lookFor: 'Item\_blank' },

        { src: 'game\_sample/js/item\_tree\_dig.js', lookFor: 'Item\_tree\_dig' },

        { src: 'game\_sample/js/item\_bush\_dig.js', lookFor: 'Item\_bush\_dig' },

        { src: 'game\_sample/js/item\_grass\_dig.js', lookFor: 'Item\_grass\_dig' },

        { src: 'game\_sample/js/item\_flower\_dig.js', lookFor: 'Item\_flower\_dig' },

        { src: 'game\_sample/js/item\_branch.js', lookFor: 'Item\_branch' },

        { src: 'game\_sample/js/character\_description.js', lookFor: 'Character\_description' },

        { src: 'game\_sample/js/synthesisBar.js', lookFor: 'SynthesisBar' },

        { src: 'game\_sample/js/equipmentBar.js', lookFor: 'EquipmentBar' },

        { src: 'game\_sample/js/clock.js', lookFor: 'Clock' },

        { src: 'game\_sample/js/backpack.js', lookFor: 'Backpack' },

        { src: 'game\_sample/js/level2\_change.js', lookFor: 'Level2\_change' },

        { src: 'game\_sample/js/handle\_initial\_character.js', lookFor: 'Handle\_initial\_character' },

        { src: 'game\_sample/js/skill\_handler.js', lookFor: 'Skill\_handler' },

        { src: 'game\_sample/js/spear\_handler.js', lookFor: 'Spear\_handler' },

        { src: 'game\_sample/js/skill\_timer.js', lookFor: 'Skill\_timer' },

        { src: 'game\_sample/js/monster\_damage\_handler.js', lookFor: 'Monster\_damage\_handler' },

        { src: 'game\_sample/js/flying\_arror.js', lookFor: 'Flying\_arror' },

        { src: 'game\_sample/js/player\_hide\_animation.js', lookFor: 'Player\_hide\_animation' },

        { src: 'game\_sample/js/proxy.js', lookFor: 'Proxy' },

        { src: 'game\_sample/js/object\_factory.js', lookFor: 'Object\_factory' },

        { src: 'game\_sample/js/weapon.js', lookFor: 'Weapon' },

        { src: 'game\_sample/js/tool.js', lookFor: 'Tool' },

        { src: 'game\_sample/js/reduceDurabilityVisitor.js', lookFor: 'ReduceDurabilityVisitor' },

        { src: 'game\_sample/js/normal\_attack.js', lookFor: 'Normal\_attack' },

        { src: 'game\_sample/js/magic\_attack.js', lookFor: 'Magic\_attack' },

        { src: 'game\_sample/js/arror\_attack.js', lookFor: 'Arror\_attack' },

        { src: 'game\_sample/js/null\_attack.js', lookFor: 'Null\_attack' },

        { src: 'game\_sample/js/handle\_game\_over.js', lookFor: 'Handle\_game\_over' },

        { src: 'game\_sample/js/monster\_base.js', lookFor: 'Monster\_base' },

        { src: 'game\_sample/js/monster\_bat.js', lookFor: 'Monster\_bat' },

        { src: 'game\_sample/js/bat\_wing.js', lookFor: 'Bat\_wing' },

        { src: 'game\_sample/js/item\_eyeball.js', lookFor: 'Item\_eyeball' },

        { src: 'game\_sample/js/monster\_pig.js', lookFor: 'Monster\_pig' },

        { src: 'game\_sample/js/monster\_cow.js', lookFor: 'Monster\_cow' },

        { src: 'game\_sample/js/monster\_bee.js', lookFor: 'Monster\_bee' },

        { src: 'game\_sample/js/monster\_cute\_little\_eye.js', lookFor: 'Monster\_cute\_little\_eye' },

        { src: 'game\_sample/js/monster\_boss.js', lookFor: 'Monster\_boss' },

        { src: 'game\_sample/js/map\_selector.js', lookFor: 'Map\_selector' },

        { src: 'game\_sample/js/local\_map\_0.js', lookFor: 'Local\_map\_0' },

        { src: 'game\_sample/js/fishing.js', lookFor: 'Fishing' },

        { src: 'game\_sample/js/item\_fish.js', lookFor: 'Item\_fish' },

        { src: 'game\_sample/js/item\_fake.js', lookFor: 'Item\_fake' },

        { src: 'game\_sample/js/npc1.js', lookFor: 'Npc1' },

        { src: 'game\_sample/js/npc2.js', lookFor: 'Npc2' },

        { src: 'game\_sample/js/npc\_event.js', lookFor: 'Npc\_event' },

        { src: 'game\_sample/js/building\_remance.js', lookFor: 'Building\_remance' },

        { src: 'game\_sample/js/creation\_blood\_status.js', lookFor: 'Creation\_blood\_status' },

        { src: 'game\_sample/js/game\_object\_detail.js', lookFor: 'Game\_object\_detail' },

        { src: 'game\_sample/js/level\_up\_animation.js', lookFor: 'Level\_up\_animation' },

        { src: 'game\_sample/js/world\_map.js', lookFor: 'World\_map' },

        { src: 'game\_sample/js/gameOver.js', lookFor: 'GameOver' },

        { src: 'game\_sample/js/mainGame.js'}

    ]

    importJS(listScript);

})();

1. Local\_map\_0.js

var Local\_map\_0 = function() {

    //91 異世界洪水  192平原 123森林 137山區 255雪地 196岩漿 200池塘

    this.houseArray = [

        //1

        [   [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

            [4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4],

        ]

    ]

    this.mapArray = [

        //1

        [ [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,255,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [255,200,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,255,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [255,200,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

        ],

        //2

        [ [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,196,196,196,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

        ],

        //3

        [ [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

        ],

        //4

        [ [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91]

        ],

        //5

        [ [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91]

        ],

.

.

.

 //100

        [   [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,196,91,91,91,91,91,91],

        [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

        [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

        [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

        [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

        [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

    ],

    ];

    this.itemArray = [];

    this.houseItemArray = [];

    this.itemArray\_two\_dimension = [];

    this.itemArray\_one\_dimension = [];

    for(var i=0;i<10;i++)

        this.itemArray.push(null);

    this.proxy = new Proxy();

    this.addObject= function(map\_name, position, number, object){

        this.itemArray[number][position.x][position.y] = object;

        switch(map\_name){

            case "World":

                this.itemArray[number][position.x][position.y] = object;

                break;

            case "House1":

                this.houseItemArray[number][position.x][position.y] = object;

                break;

            default:

                break;

        }

    }

    this.removeObject= function(map\_name, position, number){

        switch(map\_name){

            case "World":

                this.itemArray[number][position.x][position.y] = new Item\_blank();

                break;

            case "House1":

                this.houseItemArray[number][position.x][position.y] = new Item\_blank();

                break;

            default:

                break;

        }

    }

    this.hasItem= function(map\_name, position, number){

        switch(map\_name){

            case "World":

                return this.itemArray[number][position.x][position.y].item\_num == 0 ? true : false ;

            case "House1":

                return this.houseItemArray[number][position.x][position.y].item\_num == 0 ? true : false ;

            default:

                break;

        }

    }

    this.canWalk= function(map\_name, position, number){

        switch(map\_name){

            case "World":

                return this.mapArray[number][position.x][position.y] == 91 ? false : this.mapArray[number][position.x][position.y] == 200 ? false : this.mapArray[number][position.x][position.y] == 4 ? false : true;

            case "House1":

                return this.houseArray[number][position.x][position.y] == 91 ? false : this.mapArray[number][position.x][position.y] == 200 ? false : this.mapArray[number][position.x][position.y] == 4 ? false : true;

            default:

                break;

        }

    }

    this.catchMap= function(map\_name, position, number){

        switch(map\_name){

            case "World":

                return this.mapArray[number][position.x][position.y];

            case "House1":

                return this.houseArray[number][position.x][position.y];

            default:

                break;

        }

    }

    this.catchItem= function(map\_name, position, number){

        switch(map\_name){

            case "World":

                this.proxy.whichToDetect(map\_name, this.itemArray);

                this.proxy.detect(number);

                return this.itemArray[number][position.x][position.y];

            case "House1":

                this.proxy.whichToDetect(map\_name, this.houseItemArray);

                this.proxy.detect(number);

                return this.houseItemArray[number][position.x][position.y];

            default:

                break;

        }

    }

};

1. magic\_attack.js

var Magic\_attack = function(player, monster, mapPosition) {

    this.player = player;

    this.monster = monster;

    this.visitor = new ReduceDurabilityVisitor();

    this.attack = function(){

        this.attackSuccess = false;

        for(var i = 0;i < this.monster.length;i++){

            if(this.monster[i].is\_start){

                for(var j = -1;j < 2;j++){

                    for(var k = -1;k < 2;k++){

                        if(mapPosition.x+j == this.monster[i].mapPosition.x && mapPosition.y+k == this.monster[i].mapPosition.y){

                            this.monster[i].health -= this.player.character\_descruption\_total\_point[3];

                            this.monster[i].getHurt();

                        }

                    }

                }

            }

        }

        this.player.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

    }

};

1. mainGame.js

Framework.Game.fps = 60;

Framework.Game.canvasWidth = 1600;

Framework.Game.canvasHeight = 900;

Framework.Game.isBackwardCompatiable = false;

//當有要加關卡時, 可以使用addNewLevel

//第一個被加進來的Level就是啟動點, 所以一開始遊戲就進入MyMenu

Framework.Game.addNewLevel({menu: new MyMenu()});

Framework.Game.addNewLevel({drama0: new Drama0()});

Framework.Game.addNewLevel({level1: new Level2\_change()});

Framework.Game.addNewLevel({gameOver: new GameOver()});

Framework.Game.start();

Framework.Game.addNewLevel({menu\_tutorial: new MyMenu\_tutorial()});

Framework.Game.addNewLevel({menu\_gameteam: new MyMenu\_gameteam()});

1. map\_item\_tree.js

var Map\_item\_tree = function(map) {

    this.map\_item\_tree = new Framework.Sprite(define.materialPath + 'map\_item\_tree.png');

    this.map\_item\_tree.scale = 2;

    this.map\_item\_tree\_growed = new Framework.Sprite(define.materialPath + 'map\_item\_tree\_growed.png');

    this.map\_item\_tree\_growed.scale = 2;

    this.map\_item\_tree\_cutted = new Framework.Sprite(define.materialPath + 'map\_item\_tree\_cutted.png');

    this.map\_item\_tree\_cutted.scale = 2;

    this.mapPosition = {x:0, y:0};

    this.status = true;

    this.isRegenerate = true;

    this.item\_num = -1;

    //equipment: 可入裝備欄的物件 material: 可堆疊的基本物件 tool: 不可堆疊的其他物件 food: 可堆疊的食物 plant: 可堆疊植物

    this.type = "material";

    //可疊加物件有amount 不可疊加有durability

    this.amount = 1;

    //計算被按了幾次空白

    this.false\_count = 0;

    this.treeStatus = 0;

    this.dropWood = false;

    this.regeneration\_time = 5000;

    this.growing = false;

    this.grow = function(){

        setTimeout(()=>{

            if(this.treeStatus != 0){

                this.treeStatus -= 1;

                this.status = true;

                this.growing = false;

            }

        }, this.regeneration\_time);

    }

    this.tryGrow = function(){

        var interval = setInterval(()=>{

            if(!this.growing){

                this.grow();

                this.growing = true;

                clearInterval(interval);

            }

        },1000);

    }

    this.update = function(){

        this.false\_count += 1;

        if(this.false\_count == 2){

            this.treeStatus += 1;

            this.false\_count = 0;

            this.dropWood = true;

            this.tryGrow();

            if(this.treeStatus == 2)

                this.status = false;

        }else{

            this.dropWood = false;

        }

    }

    this.draw = function(ctx){

        switch(this.treeStatus){

            case 0:

                this.map\_item\_tree.draw(ctx);

                break;

            case 1:

                this.map\_item\_tree\_growed.draw(ctx);

                break;

            case 2:

                this.map\_item\_tree\_cutted.draw(ctx);

                break;

            default:

                break;

        }

    }

};

Object.defineProperty(Map\_item\_tree.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.map\_item\_tree.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

        this.map\_item\_tree\_growed.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

        this.map\_item\_tree\_cutted.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. map\_selector.js

var Map\_selector = function() {

    this.flag ={x:0, y:0};

    this.flag\_map;

    this.local\_map\_0 = new Local\_map\_0();

    this.pickObject = function(map\_name, world\_position){

        this.flag.x = Math.floor((world\_position.y) / 40);

        this.flag.y = Math.floor((world\_position.x) / 40);

        this.flag\_map = this.flag.y\*10+this.flag.x;

        this.local\_map\_0.removeObject(map\_name, {x:(world\_position.y%40), y:(world\_position.x%40)}, this.flag\_map);

    }

    this.addObject = function(map\_name, world\_position, object){

        this.flag.x = Math.floor((world\_position.y) / 40);

        this.flag.y = Math.floor((world\_position.x) / 40);

        this.flag\_map = this.flag.y\*10+this.flag.x;

        this.local\_map\_0.addObject(map\_name, {x:(world\_position.y%40), y:(world\_position.x%40)}, this.flag\_map, object);

    }

    this.checkIsBlank = function(map\_name, world\_position){

        this.flag.x = Math.floor((world\_position.y) / 40);

        this.flag.y = Math.floor((world\_position.x) / 40);

        this.flag\_map = this.flag.y\*10+this.flag.x;

        return this.local\_map\_0.hasItem(map\_name, {x:(world\_position.y%40), y:(world\_position.x%40)}, this.flag\_map);

    }

    this.checkFloorCanWalk = function(map\_name, world\_position){

        this.flag.x = Math.floor((world\_position.y) / 40);

        this.flag.y = Math.floor((world\_position.x) / 40);

        this.flag\_map = this.flag.y\*10+this.flag.x;

        return this.local\_map\_0.canWalk(map\_name, {x:(world\_position.y%40), y:(world\_position.x%40)}, this.flag\_map);

    }

    this.makeItemMap = function(map\_name, world\_position){

        this.itemArray = [];

        this.item\_line = [];

        for(var i=-5;i<6;i++){

            for(var j=-5;j<6;j++){

                this.flag.x = Math.floor((world\_position.y+i) / 40);

                this.flag.y = Math.floor((world\_position.x+j) / 40);

                this.item\_line.push(this.littleItem(map\_name, {x:((world\_position.y+i)%40), y:((world\_position.x+j)%40)}, this.flag.y\*10+this.flag.x));

            }

            this.itemArray.push(this.item\_line);

            this.item\_line = [];

        }

        return this.itemArray;

    }

    this.nullClean = function(){

        this.local\_map\_0.proxy.nullMapClean();

    }

    this.littleItem = function(map\_name, position, number){

        return this.local\_map\_0.catchItem(map\_name, position,number);

    }

    this.makeMap= function(map\_name, world\_position){

        this.mapArray = [];

        this.map\_line = [];

        for(var i=-5;i<6;i++){

            for(var j=-5;j<6;j++){

                this.flag.x = Math.floor((world\_position.y+i) / 40);

                this.flag.y = Math.floor((world\_position.x+j) / 40);

                this.map\_line.push(this.littleMap(map\_name, {x:((world\_position.y+i)%40), y:((world\_position.x+j)%40)}, this.flag.y\*10+this.flag.x));

            }

            this.mapArray.push(this.map\_line);

            this.map\_line = [];

        }

        return this.mapArray;

    }

    this.littleMap = function(map\_name, position, number){

        return this.local\_map\_0.catchMap(map\_name, position, number);

    }

};

1. monster\_base.js

var Monster\_base = function(){

    this.monster;

    this.map;

    var PIXEL\_CONST = 64;

    this.mapPosition = {x:0, y:0};

    this.walkTarget = {x:0, y:0};

    this.spritePosition = {x:0, y:0};

    this.constants = new Constants();

    this.isHurt = false;

    this.isdead = false;

    this.is\_start = false;

    this.walkStep = {x:0,y:0};

    this.isWalking = false;

    this.monsterDirection = {x:0, y:0};

    this.walkVector = {x:0, y:0};

    this.hurt = new Framework.AnimationSprite({url:define.materialPath + 'Absorb.png', col:5, row:5, loop:false, speed:12});

    this.hurt.scale = 2;

    this.init = function(map, monster, monsterDie,walkSpeed){

        this.map = map;

        this.monster = monster;

        this.walkSpeed = walkSpeed;

        this.monsterDie = monsterDie;

    }

    this.walk = function(moveStep){

        if(this.isWalking === false){

            this.isWalking = true;

            this.walkTarget = {x:this.mapPosition.x + moveStep.x, y:this.mapPosition.y + moveStep.y};

        }

    }

    this.die = function(){

        this.isHurt = true;

        this.monsterDie.start({ from: 0, to: 2, loop: false});

        setTimeout(()=>{

            this.isdead = true;

        },500);

    }

    this.getHurt = function(){

        if(!this.isHurt){

            this.isHurt = true;

            this.monsterDie.start({ from: 0, to: 2, loop: false});

            this.hurt.start({ from: 17, to: 19, loop: false});

            setTimeout(()=>{

                this.isHurt = false;

            },300);

        }

    }

    this.walkAlittle = function(){

        if(this.playerDirection === this.constants.DirectionEnum.DOWN){

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y + this.walkSpeed};

        }

        else if(this.playerDirection === this.constants.DirectionEnum.LEFT){

            this.spritePosition = {x:this.spritePosition.x - this.walkSpeed, y:this.spritePosition.y};

        }

        else if(this.playerDirection === this.constants.DirectionEnum.RIGHT){

            this.spritePosition = {x:this.spritePosition.x + this.walkSpeed, y:this.spritePosition.y};

        }

        else if(this.playerDirection === this.constants.DirectionEnum.UP){

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y - this.walkSpeed};

        }

    }

    this.update = function(){

        this.checkIsMonsterOutCanvus();

        if(!this.is\_start){ return; }

        this.hurt.update();

        this.monster.update();

        this.monsterDie.update();

        if(this.isWalking){

            if(this.walkTarget.x \* PIXEL\_CONST === this.spritePosition.x && this.walkTarget.y \* PIXEL\_CONST === this.spritePosition.y){

                this.isWalking = false;

                this.monster.index = this.playerDirection \* 3 + 1;

                this.mapPosition = this.walkTarget;

            }else{

                this.walkAlittle();

            }

        }else

        {

            if(this.is\_start)

            {

                if(this.map.player1.hide)

                    this.randomWalk();

                else{

                    this.rushToYou();

                }

            }

        }

    }

    this.draw = function(ctx){

        if(!this.is\_start){ return; }

        var xx = 13\*64 + this.spritePosition.x - this.map.playerPositionOnMap.x\*64;

        var yy = 7\*64 + this.spritePosition.y - this.map.playerPositionOnMap.y\*64;

        if(!this.isHurt){

            this.monster.position = {x: xx, y: yy};

            this.monster.draw(ctx);

        }else{

            this.monsterDie.position = {x: xx, y: yy};

            this.hurt.position = {x: xx, y: yy};

            this.monsterDie.draw(ctx);

            this.hurt.draw(ctx);

        }

    }

    this.checkIsMonsterOutCanvus = function(){

        if( Math.abs((this.mapPosition.x-this.map.playerPositionOnMap.x)) <6 &&　Math.abs((this.mapPosition.y-this.map.playerPositionOnMap.y)) <6){

            this.is\_start = true;

        }else{

            this.is\_start = false;

        }

    }

    this.howToWalk = function(restriction, thirdDirection){

        this.audio = new Framework.Audio({

            cow: {

                mp3: define.musicPath + '牛叫聲.mp3',

            },bat: {

                mp3: define.musicPath + '蝙蝠叫聲.mp3',

            },pig: {

                mp3: define.musicPath + '豬叫聲.mp3',

            },eye: {

                mp3: define.musicPath + '眼球叫聲.mp3',

            }

        });

        switch(this.name){

            case "閃耀魔眼":

                this.audio.play({name: 'cow', loop: false});

                break;

            case "小蝙蝠":

                this.audio.play({name: 'bat', loop: false});

                break;

            case "小豬":

                this.audio.play({name: 'pig', loop: false});

                break;

            case "大眼仔仔":

                this.audio.play({name: 'eye', loop: false});

                break;

            default:

                break;

        }

        var directionArray = [{x:-1,y:0},{x:1,y:0},{x:0,y:-1},{x:0,y:1}];

        var thirdDirection = directionArray[thirdDirection];

        directionArray.splice( restriction, 1 );

        for(var i=0;i<directionArray.length;i++){

            if(directionArray[i].x == thirdDirection.x && directionArray[i].y == thirdDirection.y){

                directionArray.splice( i, 1 );

                break;

            }

        }

        var randum\_number = Math.floor(Math.random()\*2) ;

        if(this.map.checkMonsterIsWalkAble({x: this.mapPosition.x +directionArray[randum\_number].x,y:this.mapPosition.y + directionArray[randum\_number].y}))

        {

            return directionArray[randum\_number];

        }else

        {

            directionArray.splice( randum\_number, 1 );

            if(this.map.checkMonsterIsWalkAble({x: this.mapPosition.x +directionArray[0].x, y:this.mapPosition.y + directionArray[0].y}))

            {

                return directionArray[0];

            }else{

                if(this.map.checkMonsterIsWalkAble({x:this.mapPosition.x +thirdDirection.x, y:this.mapPosition.y +thirdDirection.y}))

                {

                    return thirdDirection;

                }else{

                    return {x:0, y:0};

                }

            }

        }

    }

    this.isAttack = function(){

        if(this.monsterDirection.x + this.mapPosition.x == this.map.playerPositionOnMap.x && this.monsterDirection.y + this.mapPosition.y == this.map.playerPositionOnMap.y && !this.isHurt){

            return true;

        }

        return false;

    }

    this.rushToYou = function()

    {

        var walkStep = {x:0,y:0};

        this.walkVector = {x:this.mapPosition.x-this.map.playerPositionOnMap.x, y:this.mapPosition.y-this.map.playerPositionOnMap.y};

        if(this.walkVector.x == 0){

            walkStep = this.walkVector.y > 0 ? {x:0,y:-1} : {x:0,y:1};

        }else if(this.walkVector.y == 0){

            walkStep = this.walkVector.x > 0 ? {x:-1,y:0} : {x:1,y:0};

        }else if( Math.abs(this.walkVector.x) >= Math.abs(this.walkVector.y)){

            if(this.walkVector.x < 0){

                if(this.walkVector.y>0)

                    walkStep = this.howToWalk(0,3);

                else

                    walkStep = this.howToWalk(0,2);

            }else{

                if(this.walkVector.y>0)

                    walkStep = this.howToWalk(1,3);

                else

                    walkStep = this.howToWalk(1,2);

            }

        }else if(Math.abs(this.walkVector.x) < Math.abs(this.walkVector.y)){

            if(this.walkVector.y < 0){

                if(this.walkVector.x>0)

                    walkStep = this.howToWalk(2,1);

                else

                    walkStep = this.howToWalk(2,0);

            }else{

                if(this.walkVector.x>0)

                    walkStep = this.howToWalk(3,1);

                else

                    walkStep = this.howToWalk(3,0);

            }

        }else{

            this.changeWalkDirection(this.walkStep);

            return;

        }

        if(this.walkStep.x != walkStep.x || this.walkStep.y != walkStep.y)

            this.changeWalkDirection(walkStep);

        if(this.map.checkMonsterIsWalkAble({x: this.mapPosition.x + walkStep.x, y:this.mapPosition.y + walkStep.y}))

        {

            this.walk(walkStep);

        }

    }

    this.changeWalkDirection = function(walkStep){

        this.walkStep = walkStep;

        if(walkStep.x > 0){

            this.playerDirection = this.constants.DirectionEnum.RIGHT;

            this.monsterDirection = this.constants.Direction.RIGHT;

        }else if(walkStep.x <0){

            this.playerDirection = this.constants.DirectionEnum.LEFT;

            this.monsterDirection = this.constants.Direction.LEFT;

        }

        if(walkStep.y > 0){

            this.playerDirection = this.constants.DirectionEnum.DOWN;

            this.monsterDirection = this.constants.Direction.DOWN;

        }else if(walkStep.y < 0){

            this.playerDirection = this.constants.DirectionEnum.UP;

            this.monsterDirection = this.constants.Direction.UP;

        }

        this.monster.start({ from: this.playerDirection \* 3, to: this.playerDirection \* 3 + 2, loop: true});

    }

    this.randomWalk = function()

    {

        var randNum = Framework.Game.\_currentLevel.cycleCount % 553;

        var walkStep = {x:0,y:0};

        if(randNum % 117 == 0)

        {

            walkStep.x = 1

        }else if(randNum % 79 == 0)

        {

            walkStep.x = -1

        }else if(randNum % 133 == 0)

        {

            walkStep.y = 1

        }else if(randNum % 157 == 0)

        {

            walkStep.y = -1

        }else

        {

            return;

        }

        if(this.walkStep.x != walkStep.x || this.walkStep.y != walkStep.y)

            this.changeWalkDirection(walkStep);

        if(this.map.checkMonsterIsWalkAble( {x: this.mapPosition.x + walkStep.x, y:this.mapPosition.y + walkStep.y} ))

        {

            this.walk(walkStep);

        }

    }

}

Object.defineProperty(Monster\_base.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.spritePosition = {x:this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

Object.defineProperty(Monster\_base.prototype, 'isDead', {

    get: function() {

        return this.isdead;

    }

});

1. monster\_bat.js

var Monster\_bat = function(map) {

    this.monster\_bat = new Framework.AnimationSprite({url:define.materialPath + 'bat.png', col:3 , row:4 , loop:true , speed:12});

    this.monster\_bat.scale = 1.3;

    this.monster\_bat.index = 1;

    this.monster\_bat\_die = new Framework.AnimationSprite({url:define.materialPath + 'bat\_die.png', col:3 , row:1 , loop:false , speed:12});

    this.monster\_bat\_die.scale = 1.2;

    this.monster\_tocan = new Framework.Sprite(define.materialPath + 'bat\_tocan.png');

    this.monster\_tocan.scale = 0.8;

    this.name = "小蝙蝠";

    this.attack = 80;

    this.health = 1000;

    this.maxHealth = 1000;

    //地圖，圖片，walkSpeed

    this.init(map, this.monster\_bat, this.monster\_bat\_die, 8);

    this.drop = function(){

        var random = Math.floor(Math.random()\*2);

        if(random == 0)

            return new Bat\_wing();

        if(random == 1)

            return new Item\_monster\_meat();

    }

};

Monster\_bat.prototype = new Monster\_base();

1. monster\_bee.js

var Monster\_bee = function(map) {

    this.monster\_bee = new Framework.AnimationSprite({url:define.materialPath + 'bee.png', col:3 , row:4 , loop:true , speed:12});

    this.monster\_bee.scale = 1.2;

    this.monster\_bee.index = 1;

    this.monster\_bee\_die = new Framework.AnimationSprite({url:define.materialPath + 'bee.png', col:3 , row:4 , loop:false , speed:12});

    this.monster\_bee\_die.scale = 1.2;

    this.monster\_tocan = new Framework.Sprite(define.materialPath + 'bee\_tocan.png');

    this.monster\_tocan.scale = 0.8;

    this.name = "蜂哥";

    this.attack = 60;

    this.health = 200;

    this.maxHealth = 200;

    //地圖，圖片，walkSpeed

    this.init(map, this.monster\_bee, this.monster\_bee\_die, 8);

    this.drop = function(){

        var random = Math.floor(Math.random()\*2);

        if(random == 0)

            return new Item\_bee\_sting();

        if(random == 1)

            return new Item\_honey();

    }

};

Monster\_bee.prototype = new Monster\_base();

1. monster\_boss.js

var Monster\_boss = function(map) {

    this.monster\_boss = new Framework.AnimationSprite({url:define.materialPath + 'boss.png', col:3 , row:4 , loop:true , speed:12});

    this.monster\_boss.scale = 1.3;

    this.monster\_boss.index = 1;

    this.monster\_boss\_die = new Framework.AnimationSprite({url:define.materialPath + 'boss\_die.png', col:3 , row:1 , loop:false , speed:12});

    this.monster\_boss\_die.scale = 1.2;

    this.monster\_tocan = new Framework.Sprite(define.materialPath + 'boss\_tocan.png');

    this.monster\_tocan.scale = 0.8;

    this.normal\_attack = new Framework.AnimationSprite({url:define.skillAnimationPath + 'ClawSpecial1.png', col:5 , row:5 , loop:false , speed:12});

    this.normal\_attack.position = {x:13\*64,y:7\*64};

    this.normal\_attack.scale = 0.8;

    this.hurtRevolution =  new Framework.AnimationSprite({url:define.skillAnimationPath + 'Darkness1.png', col:5 , row:4 , loop:false , speed:12});

    this.hurtRevolution.scale = 2;

    this.transport = new Framework.AnimationSprite({url:define.skillAnimationPath + 'Thunder3.png', col:5 , row:1 , loop:false , speed:12});

    this.transport.position = {x:13\*64,y:7\*64};

    this.transport.scale = 0.8;

    this.range\_attack = new Framework.AnimationSprite({url:define.skillAnimationPath + 'Darkness5.png', col:5 , row:5 , loop:false , speed:12});

    this.range\_attack.scale = 2;

    this.rangePosition = {x:0,y:0};

    this.remote\_attack = new Framework.AnimationSprite({url:define.skillAnimationPath + 'Darkness4.png', col:5 , row:6 , loop:false , speed:12});

    this.remote\_attack.scale = 0.8;

    this.remotePosition = {x:0,y:0};

    this.remoteSprite = {x:0,y:0};

    this.remoteTarget = {x:0,y:0};

    this.remoteDirection = {x:0,y:0};

    this.speed = 16;

    this.catch = [];

    this.catchPosition = [];

    for(var i=0;i < 4;i++){

        var img = new Framework.AnimationSprite({url:define.skillAnimationPath + 'StateDown2.png', col:5 , row:5 , loop:false , speed:12});

        img.scale = 0.4;

        this.catch.push(img);

    }

    this.name = "boss";

    this.attack = 150;

    this.health = 6000;

    this.maxHealth = 6000;

    this.monster\_boss;

    this.map = map;

    var PIXEL\_CONST = 64;

    this.mapPosition = {x:0, y:0};

    this.walkTarget = {x:0, y:0};

    this.spritePosition = {x:0, y:0};

    this.constants = new Constants();

    this.isHurt = false;

    this.isdead = false;

    this.is\_start = false;

    this.walkStep = {x:0,y:0};

    this.isWalking = false;

    this.monsterDirection = {x:0, y:0};

    this.walkVector = {x:0, y:0};

    this.hurt = new Framework.AnimationSprite({url:define.materialPath + 'Absorb.png', col:5, row:5, loop:false, speed:12});

    this.hurt.scale = 2;

    this.walkSpeed = 8;

    this.attackTimer = 0;

    this.trace = false;

    this.isCatching = false;

    this.frequency = 15;

    this.catchPlayer = function(){

        var audio = new Framework.Audio({

            catch: {

                mp3: define.musicPath + '綑綁.mp3'

            }

        });

        audio.play({name: 'catch', loop: false});

        this.catchPosition = [];

        this.catchPosition.push({x:this.map.playerPositionOnMap.x+1, y:this.map.playerPositionOnMap.y});

        this.catchPosition.push({x:this.map.playerPositionOnMap.x-1, y:this.map.playerPositionOnMap.y});

        this.catchPosition.push({x:this.map.playerPositionOnMap.x, y:this.map.playerPositionOnMap.y-1});

        this.catchPosition.push({x:this.map.playerPositionOnMap.x, y:this.map.playerPositionOnMap.y+1});

        for(var i = 0;i < 4;i++){

            this.catch[i].start({from:11,to:24,loop:true});

        }

        this.isCatching = true;

        setTimeout(()=>{

            if(!this.map.player1.beCaught){

                this.isCatching = false;

                for(var i = 0;i < 4;i++){

                    this.catch[i].stop();

                }

            }

        },1500);

        var interval = setInterval(()=>{

            if(!this.map.player1.beCaught && this.isCatching)

                this.checkIfCatch();

            if(!this.isCatching || this.map.player1.beCaught)

                clearInterval(interval);

        },5)

    }

    this.checkIfCatch = function(){

        for(var i = 0;i < 4;i++){

            if(this.catchPosition[i].x == this.map.playerPositionOnMap.x && this.catchPosition[i].y == this.map.playerPositionOnMap.y){

                this.map.player1.beCaught = true;

                for(var j = 0;j < 4;j++){

                    if(i == j)

                        continue;

                    this.catch[j].stop();

                }

                setTimeout(()=>{

                    this.isCatching = false;

                    this.map.player1.beCaught = false;

                    this.catch[i].stop();

                },3000);

                break;

            }

        }

    }

    this.drop = function(){

        var random = Math.floor(Math.random()\*2);

        if(random == 0)

            return new Bat\_wing();

        if(random == 1)

            return new Item\_monster\_meat();

    }

    this.walk = function(moveStep){

        if(this.isWalking === false){

            this.isWalking = true;

            this.walkTarget = {x:this.mapPosition.x + moveStep.x, y:this.mapPosition.y + moveStep.y};

        }

    }

    this.die = function(){

        this.isHurt = true;

        this.monster\_boss\_die.start({ from: 0, to: 2, loop: false});

        setTimeout(()=>{

            this.isdead = true;

        },500);

    }

    this.getHurt = function(){

        if(!this.isHurt){

            this.isHurt = true;

            this.monster\_boss\_die.start({ from: 0, to: 2, loop: false});

            this.hurt.start({ from: 17, to: 19, loop: false});

            if(!this.hurtRevolution.\_start)

                this.revolution();

            setTimeout(()=>{

                this.isHurt = false;

            },200);

        }

    }

    this.revolution = function(){

        var random = Math.floor(Math.random()\*5);

        if(random == 1)

            this.hurtRevolution.start({ from: 17, to: 19, loop: false});

        this.map.player1.gethurt(120);

    }

    this.walkAlittle = function(){

        if(this.playerDirection === this.constants.DirectionEnum.DOWN)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y + this.walkSpeed};

        else if(this.playerDirection === this.constants.DirectionEnum.LEFT)

            this.spritePosition = {x:this.spritePosition.x - this.walkSpeed, y:this.spritePosition.y};

        else if(this.playerDirection === this.constants.DirectionEnum.RIGHT)

            this.spritePosition = {x:this.spritePosition.x + this.walkSpeed, y:this.spritePosition.y};

        else if(this.playerDirection === this.constants.DirectionEnum.UP)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y - this.walkSpeed};

    }

    this.update = function(){

        if(this.isdead){ return; }

        this.checkIsMonsterOutCanvus();

        if(!this.is\_start){ return; }

        if(this.health <= 0)

            this.die();

        this.normal\_attack.update();

        this.hurt.update();

        this.monster\_boss.update();

        this.monster\_boss\_die.update();

        this.transport.update();

        this.hurtRevolution.update();

        this.range\_attack.update();

        this.remote\_attack.update();

        for(var i = 0;i < 4;i++)

            this.catch[i].update();

        this.attackTimer += 1;

        if(this.remote\_attack.\_start){

            this.remoteSprite.x += this.speed\*this.remoteDirection.x;

            this.remoteSprite.y += this.speed\*this.remoteDirection.y;

            if(this.remoteSprite.x == this.remoteTarget.x\*64 && this.remoteSprite.y == this.remoteTarget.y\*64){

                this.remotePosition.x = this.remoteTarget.x;

                this.remotePosition.y = this.remoteTarget.y;

                this.remoteTarget.x += this.remoteDirection.x;

                this.remoteTarget.y += this.remoteDirection.x;

            }

            if(this.remotePosition.x == this.map.playerPositionOnMap.x && this.remotePosition.y == this.map.playerPositionOnMap.y){

                this.map.player1.gethurt(150);

                this.remote\_attack.stop();

            }

        }

        if(this.trace && this.attackTimer == this.frequency){

            this.transportToPlayer();

        }else{

            if(this.isWalking){

                if(this.walkTarget.x \* PIXEL\_CONST === this.spritePosition.x && this.walkTarget.y \* PIXEL\_CONST === this.spritePosition.y){

                    var audio = new Framework.Audio({

                        roar: {

                            mp3: define.musicPath + '魔王叫聲.mp3',

                        }

                    });

                    this.isWalking = false;

                    this.monster\_boss.index = this.playerDirection \* 3 + 1;

                    this.mapPosition = this.walkTarget;

                }else{

                    this.walkAlittle();

                }

            }else

            {

                if(this.map.player1.hide)

                    this.randomWalk();

                else{

                    this.rushToYou();

                }

            }

        }

        if(this.attackTimer == this.frequency)

            this.operationChoose();

        if(this.isAttack() && this.attackTimer == this.frequency && !this.normal\_attack.\_start){

            this.normal\_attack.start({ from: 0, to: 5, loop: false});

        }

        if(this.attackTimer == this.frequency)

            this.attackTimer = 0;

    }

    this.operationChoose = function(){

        var random = Math.floor(Math.random()\*10);

        if(!this.isCatching && random == 1)

            this.catchPlayer();

        if(!this.range\_attack.\_start && random == 2)

            this.rangeAttack();

        if(!this.remote\_attack.\_start && random == 3)

            this.remoteAttack();

    }

    this.remoteAttack = function(){

        var audio1 = new Framework.Audio({

            drop: {

                mp3: define.musicPath + '摔東西.mp3'

            }

        });

        audio1.play({name: 'drop', loop: false});

        this.remote\_attack.start({from:18,to:25,loop:false});

        this.remotePosition.x = this.mapPosition.x+this.monsterDirection.x;

        this.remotePosition.y = this.mapPosition.y+this.monsterDirection.y;

        this.remoteDirection.x = this.monsterDirection.x;

        this.remoteDirection.y = this.monsterDirection.y;

        this.remoteSprite = {x:this.remotePosition.x\*64,y:this.remotePosition.y\*64};

        this.remoteTarget.x = this.remotePosition.x + this.remoteDirection.x;

        this.remoteTarget.y = this.remotePosition.y + this.remoteDirection.y;

        var interval = setInterval(()=>{

            if(this.remote\_attack.\_start){

                this.remotePosition.x += this.remoteDirection.x;

                this.remotePosition.y += this.remoteDirection.y;

            }else{

                clearInterval(interval);

            }

        },300)

    }

    this.rangeAttack = function(){

        var audio = new Framework.Audio({

            light: {

                mp3: define.musicPath + '範圍攻擊.mp3'

            }

        });

        audio.play({name: 'light', loop: false});

        this.range\_attack.start({from:1,to:24,loop:false});

        this.rangePosition.x = this.map.playerPositionOnMap.x;

        this.rangePosition.y = this.map.playerPositionOnMap.y;

        setTimeout(()=>{

            this.checkIfRangeAttackHit();

        },200);

    }

    this.checkIfRangeAttackHit = function(){

        var interval = setInterval(()=>{

            if(this.range\_attack.\_start){

                for(var i = -1;i < 2;i++){

                    for(var j = -1;j < 2;j++){

                        if(this.map.playerPositionOnMap.x == this.rangePosition.x + i && this.map.playerPositionOnMap.y == this.rangePosition.y + j)

                            this.map.player1.gethurt(120);

                    }

                }

            }else{

                clearInterval(interval);

            }

        },500)

    }

    this.draw = function(ctx){

        if(this.isdead){ return; }

        if(!this.is\_start){ return; }

        if(!( Math.abs((this.mapPosition.x-this.map.playerPositionOnMap.x)) <6 &&　Math.abs((this.mapPosition.y-this.map.playerPositionOnMap.y)) <6)){return;}

        var xx = 13\*64 + this.spritePosition.x - this.map.playerPositionOnMap.x\*64;

        var yy = 7\*64 + this.spritePosition.y - this.map.playerPositionOnMap.y\*64;

        if(this.isCatching){

            for(var i = 0;i < 4;i++){

                var x1 = 13\*64 + this.catchPosition[i].x\*64 - this.map.playerPositionOnMap.x\*64;

                var y1 = 7\*64 + this.catchPosition[i].y\*64 - this.map.playerPositionOnMap.y\*64;

                if(this.catch[i].\_start && this.checkIfDraw({x:this.catchPosition[i].x\*64,y:this.catchPosition[i].y\*64})){

                    this.catch[i].position = {x:x1,y:y1};

                    this.catch[i].draw(ctx);

                }

            }

        }

        if(this.range\_attack.\_start){

            var x2 = 13\*64 + this.rangePosition.x\*64 - this.map.playerPositionOnMap.x\*64;

            var y2 = 7\*64 + this.rangePosition.y\*64 - this.map.playerPositionOnMap.y\*64;

            if(this.checkIfDraw({x:this.rangePosition.x\*64,y:this.rangePosition.y\*64})){

                this.range\_attack.position = {x:x2,y:y2};

                this.range\_attack.draw(ctx);

            }

        }

        if(this.remote\_attack.\_start){

            var x3 = 13\*64 + this.remoteSprite.x - this.map.playerPositionOnMap.x\*64;

            var y3 = 7\*64 + this.remoteSprite.y - this.map.playerPositionOnMap.y\*64;

            if(this.checkIfDraw(this.remoteSprite)){

                this.remote\_attack.position = {x:x3,y:y3};

                this.remote\_attack.draw(ctx);

            }

        }

        if(this.hurtRevolution.\_start){

            this.hurtRevolution.position = {x:xx,y:yy};

            this.hurtRevolution.draw(ctx);

        }

        if(this.normal\_attack.\_start && this.isAttack())

            this.normal\_attack.draw(ctx);

        if(this.transport.\_start){

            this.transport.position = {x: xx, y: yy};

            this.transport.draw(ctx);

        }else{

            if(!this.isHurt){

                this.monster\_boss.position = {x: xx, y: yy};

                this.monster\_boss.draw(ctx);

            }else{

                this.monster\_boss\_die.position = {x: xx, y: yy};

                this.hurt.position = {x: xx, y: yy};

                this.monster\_boss\_die.draw(ctx);

                this.hurt.draw(ctx);

            }

        }

    }

    this.checkIsMonsterOutCanvus = function(){

        if( Math.abs((this.mapPosition.x-this.map.playerPositionOnMap.x)) <6 &&　Math.abs((this.mapPosition.y-this.map.playerPositionOnMap.y)) <6){

            this.is\_start = true;

            this.trace = false;

        }else

            this.trace = true;

    }

    this.checkIfDraw = function(sprite){

        if( Math.abs((sprite.x-this.map.playerPositionOnMap.x\*64)) <6\*64 &&　Math.abs((sprite.y-this.map.playerPositionOnMap.y\*64)) <6\*64)

            return true;

        else

            return false;

    }

    this.transportToPlayer = function(){

        var i = (Math.floor(Math.random()\*3) + 1)\*(Math.random() > 0.5 ? 1 : -1);

        var j = (Math.floor(Math.random()\*3) + 1)\*(Math.random() > 0.5 ? 1 : -1);

        var audio1 = new Framework.Audio({

            move: {

                mp3: define.musicPath + '順移.mp3'

            }

        });

        audio1.play({name: 'move', loop: false});

        if(this.map.mapArray[5+i][5+j] != 91 &&

            this.map.mapArray[5+i][5+j] != 200 &&

            this.map.itemArray[5+i][5+j].item\_num == 0

        ){

            this.transport.start({ from: 0, to: 4, loop: false });

            this.mapPosition = {x:this.map.playerPositionOnMap.x+j,y:this.map.playerPositionOnMap.y+i};

            this.spritePosition = {x:this.mapPosition.x\*64, y:this.mapPosition.y\*64};

            this.isWalking = false;

        }

        else

            this.transportToPlayer();

    }

    this.howToWalk = function(restriction, thirdDirection){

        var directionArray = [{x:-1,y:0},{x:1,y:0},{x:0,y:-1},{x:0,y:1}];

        var thirdDirection = directionArray[thirdDirection];

        directionArray.splice( restriction, 1 );

        for(var i=0;i<directionArray.length;i++){

            if(directionArray[i].x == thirdDirection.x && directionArray[i].y == thirdDirection.y){

                directionArray.splice( i, 1 );

                break;

            }

        }

        var randum\_number = Math.floor(Math.random()\*2) ;

        if(this.map.checkMonsterIsWalkAble({x: this.mapPosition.x +directionArray[randum\_number].x,y:this.mapPosition.y + directionArray[randum\_number].y}))

        {

            return directionArray[randum\_number];

        }

        else

        {

            directionArray.splice( randum\_number, 1 );

            if(this.map.checkMonsterIsWalkAble({x: this.mapPosition.x +directionArray[0].x, y:this.mapPosition.y + directionArray[0].y}))

            {

                return directionArray[0];

            }else{

                if(this.map.checkMonsterIsWalkAble({x:this.mapPosition.x +thirdDirection.x, y:this.mapPosition.y +thirdDirection.y}))

                    return thirdDirection;

                else

                    return {x:0, y:0};

            }

        }

    }

    this.isAttack = function(){

        if(this.monsterDirection.x + this.mapPosition.x == this.map.playerPositionOnMap.x && this.monsterDirection.y + this.mapPosition.y == this.map.playerPositionOnMap.y && !this.isHurt)

            return true;

        return false;

    }

    this.rushToYou = function()

    {

        var walkStep = {x:0,y:0};

        this.walkVector = {x:this.mapPosition.x-this.map.playerPositionOnMap.x, y:this.mapPosition.y-this.map.playerPositionOnMap.y};

        if(this.walkVector.x == 0){

            walkStep = this.walkVector.y > 0 ? {x:0,y:-1} : {x:0,y:1};

        }else if(this.walkVector.y == 0){

            walkStep = this.walkVector.x > 0 ? {x:-1,y:0} : {x:1,y:0};

        }else if( Math.abs(this.walkVector.x) >= Math.abs(this.walkVector.y)){

            if(this.walkVector.x < 0){

                if(this.walkVector.y>0)

                    walkStep = this.howToWalk(0,3);

                else

                    walkStep = this.howToWalk(0,2);

            }else{

                if(this.walkVector.y>0)

                    walkStep = this.howToWalk(1,3);

                else

                    walkStep = this.howToWalk(1,2);

            }

        }else if(Math.abs(this.walkVector.x) < Math.abs(this.walkVector.y)){

            if(this.walkVector.y < 0){

                if(this.walkVector.x>0)

                    walkStep = this.howToWalk(2,1);

                else

                    walkStep = this.howToWalk(2,0);

            }else{

                if(this.walkVector.x>0)

                    walkStep = this.howToWalk(3,1);

                else

                    walkStep = this.howToWalk(3,0);

            }

        }else{

            this.changeWalkDirection(this.walkStep);

            return;

        }

        if(this.walkStep.x != walkStep.x || this.walkStep.y != walkStep.y)

            this.changeWalkDirection(walkStep);

        if(this.map.checkMonsterIsWalkAble({x: this.mapPosition.x + walkStep.x, y:this.mapPosition.y + walkStep.y}))

            this.walk(walkStep);

    }

    this.changeWalkDirection = function(walkStep){

        this.walkStep = walkStep;

        if(walkStep.x > 0){

            this.playerDirection = this.constants.DirectionEnum.RIGHT;

            this.monsterDirection = this.constants.Direction.RIGHT;

        }else if(walkStep.x <0){

            this.playerDirection = this.constants.DirectionEnum.LEFT;

            this.monsterDirection = this.constants.Direction.LEFT;

        }

        if(walkStep.y > 0){

            this.playerDirection = this.constants.DirectionEnum.DOWN;

            this.monsterDirection = this.constants.Direction.DOWN;

        }else if(walkStep.y < 0){

            this.playerDirection = this.constants.DirectionEnum.UP;

            this.monsterDirection = this.constants.Direction.UP;

        }

        this.monster\_boss.start({ from: this.playerDirection \* 3, to: this.playerDirection \* 3 + 2, loop: true});

    }

    this.randomWalk = function()

    {

        var randNum = Framework.Game.\_currentLevel.cycleCount % 553;

        var walkStep = {x:0,y:0};

        if(randNum % 117 == 0)

            walkStep.x = 1

        else if(randNum % 79 == 0)

            walkStep.x = -1

        else if(randNum % 133 == 0)

            walkStep.y = 1

        else if(randNum % 157 == 0)

            walkStep.y = -1

        else

            return;

        if(this.walkStep.x != walkStep.x || this.walkStep.y != walkStep.y)

            this.changeWalkDirection(walkStep);

        if(this.map.checkMonsterIsWalkAble( {x: this.mapPosition.x + walkStep.x, y:this.mapPosition.y + walkStep.y} ))

            this.walk(walkStep);

    }

};

Object.defineProperty(Monster\_boss.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.spritePosition = {x:this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

Object.defineProperty(Monster\_boss.prototype, 'isDead', {

    get: function() {

        return this.isdead;

    }

});

1. monster\_cow.js

var Monster\_cow = function(map) {

    this.monster\_cow = new Framework.AnimationSprite({url:define.materialPath + 'cow.png', col:3 , row:4 , loop:true , speed:12});

    this.monster\_cow.scale = 1.3;

    this.monster\_cow.index = 1;

    this.monster\_cow\_die = new Framework.AnimationSprite({url:define.materialPath + 'cow\_die.png', col:3 , row:1 , loop:false , speed:12});

    this.monster\_cow\_die.scale = 1.2;

    this.monster\_tocan = new Framework.Sprite(define.materialPath + 'cow\_tocan.png');

    this.monster\_tocan.scale = 0.8;

    this.name = "閃耀魔眼";

    this.attack = 100;

    this.health = 200;

    this.maxHealth = 200;

    //地圖，圖片，walkSpeed

    this.init(map, this.monster\_cow, this.monster\_cow\_die, 8);

    this.drop = function(){

        var random = Math.floor(Math.random()\*2);

        if(random == 0)

            return new Item\_meat();

        if(random == 1)

            return new Item\_spider\_web();

    }

};

Monster\_cow.prototype = new Monster\_base();

1. monster\_cute\_little\_eye.js

var Monster\_cute\_little\_eye = function(map) {

    this.monster\_cute\_little\_eye = new Framework.AnimationSprite({url:define.materialPath + 'monster\_cute\_litter\_eye.png', col:3 , row:4 , loop:true , speed:12});

    this.monster\_cute\_little\_eye.scale = 1.5;

    this.monster\_cute\_little\_eye.index = 1;

    this.monster\_cute\_little\_eye\_die = new Framework.AnimationSprite({url:define.materialPath + 'eye\_die.png', col:3 , row:1 , loop:true , speed:12});

    this.monster\_cute\_little\_eye\_die.scale = 1.2;

    this.monster\_tocan = new Framework.Sprite(define.materialPath + 'monster\_cute\_litter\_eye\_tocan.png');

    this.monster\_tocan.scale = 1.2;

    this.name = "大眼仔仔";

    this.attack = 55;

    this.health = 100;

    this.maxHealth = 100;

    //地圖，圖片，walkSpeed

    this.init(map, this.monster\_cute\_little\_eye, this.monster\_cute\_little\_eye\_die, 8);

    this.drop = function(){

        var random = Math.floor(Math.random()\*2);

        if(random == 0)

            return new Item\_eyeball();

        if(random == 1)

            return new Item\_droplet();

    }

};

Monster\_cute\_little\_eye.prototype = new Monster\_base();

1. monster\_pig.js

var Monster\_pig = function(map) {

    this.monster\_pig = new Framework.AnimationSprite({url:define.materialPath + 'pig.png', col:3 , row:4 , loop:true , speed:12});

    this.monster\_pig.scale = 1.3;

    this.monster\_pig.index = 1;

    this.monster\_pig\_die = new Framework.AnimationSprite({url:define.materialPath + 'pig\_die.png', col:3 , row:1 , loop:false , speed:12});

    this.monster\_pig\_die.scale = 1.2;

    this.monster\_tocan = new Framework.Sprite(define.materialPath + 'pig\_tocan.png');

    this.monster\_tocan.scale = 0.8;

    this.name = "小豬";

    this.attack =100;

    this.health = 200;

    this.maxHealth = 200;

    //地圖，圖片，walkSpeed

    this.init(map, this.monster\_pig, this.monster\_pig\_die, 8);

    this.drop = function(){

        var random = Math.floor(Math.random()\*2);

        if(random == 0)

            return new Item\_pigskin();

        if(random == 1)

            return new Item\_meat();

    }

};

Monster\_pig.prototype = new Monster\_base();

1. myMenu.js

var MyMenu = Framework.exClass(Framework.GameMainMenu , {

    //初始化loadingProgress需要用到的圖片

    initializeProgressResource: function() {

        this.loading = new Framework.Sprite(define.imagePath + 'loading.jpg');

        this.loading.position = {x: Framework.Game.getCanvasWidth() / 2 , y: Framework.Game.getCanvasHeight() / 2};

    },

    //在initialize時會觸發的事件

    loadingProgress: function(ctx, requestInfo) {

        this.loading.draw(ctx);

        ctx.font ='90px Arial';

        ctx.textAlign = 'center';

        ctx.fillStyle = 'white';

        ctx.fillText(Math.round(requestInfo.percent) + '%' , ctx.canvas.width / 2 , ctx.canvas.height / 2 + 300);

    },

    load: function() {

        this.music\_start = false;

        this.menu = new Framework.Sprite(define.imagePath + 'game\_loading.png');

        this.botton\_start = new Framework.Sprite(define.welcomimgPath + 'start\_game\_untouch.png');

        this.botton\_tutorial = new Framework.Sprite(define.welcomimgPath + 'tutorial\_game\_untouch.png');

        this.botton\_quit = new Framework.Sprite(define.welcomimgPath + 'quit\_game\_untouch.png');

        this.botton\_team = new Framework.Sprite(define.welcomimgPath + 'team\_game\_untouch.png');

        this.tutorial\_game\_touch = new Framework.Sprite(define.welcomimgPath + 'tutorial\_game\_touch.png');

        this.start\_game\_touch = new Framework.Sprite(define.welcomimgPath + 'start\_game\_touch.png');

        this.team\_game\_touch = new Framework.Sprite(define.welcomimgPath + 'team\_game\_touch.png');

        this.quit\_game\_touch = new Framework.Sprite(define.welcomimgPath + 'quit\_game\_touch.png');

    },

    initialize: function() {

        this.menu.position = {

            x: Framework.Game.getCanvasWidth() / 2,

            y: Framework.Game.getCanvasHeight() / 2

        };

        this.menu.scale = 0.9;

        this.rectPosition = {

            x: Framework.Game.getCanvasWidth() / 2 - 130,

            y: 0

        };

        this.botton\_start.position = {

            x: 1050+196,

            y: 350

        };

        this.botton\_start.scale = 1;

        this.botton\_tutorial.position = {

            x: 1050+196,

            y: 460

        };

        this.botton\_tutorial.scale = 1;

        this.botton\_quit.position = {

            x: 1050+196,

            y: 570

        };

        this.botton\_quit.scale = 1;

        this.botton\_team.position = {

            x: 1050+196,

            y: 680

        };

        this.botton\_team.scale = 1;

        this.audio = new Framework.Audio({

            start\_game: {

                mp3: define.musicPath + '遊戲開場音樂.mp3',

            }, keyup:{

                mp3: define.musicPath + '按按鍵.mp3',

            }, song2:{

                mp3: define.musicPath + '刀劍神域op1《crossing field》鋼琴曲.mp3',

            }

        });

        this.audio.play({name: 'start\_game', loop: true});

    },

    draw: function(parentCtx) {

        this.menu.draw(parentCtx);

        this.botton\_start.draw(parentCtx);

        this.botton\_tutorial.draw(parentCtx);

        this.botton\_quit.draw(parentCtx);

        this.botton\_team.draw(parentCtx);

    },

    click:function(e){

        this.audio.play({name: 'keyup', loop: false});

        if(e.x<1300+196 && e.x>1058+196 && e.y<465 && e.y>365){

            this.audio.stop('start\_game');

            Framework.Game.goToLevel('drama0');

        }

        if(e.x<1300+196 && e.x>1058+196 && e.y<577 && e.y>480){

            this.audio.stop('start\_game');

            Framework.Game.goToLevel('menu\_tutorial');

        }

        if(e.x<1300+196 && e.x>1058+196 && e.y<686 && e.y>588){

            this.audio.stop('start\_game');

            window.close();

        }

        if(e.x<1300+196 && e.x>1058+196 && e.y<798 && e.y>699){

            this.audio.stop('start\_game');

            Framework.Game.goToLevel('menu\_gameteam');

        }

    },

    mousemove: function(e) {

        if(e.x >= 1060+196 && e.x<=1296+196 && e.y<=466 && e.y>=369)

            this.botton\_start = new Framework.Sprite(define.welcomimgPath + 'start\_game\_touch.png');

        else

            this.botton\_start = new Framework.Sprite(define.welcomimgPath + 'start\_game\_untouch.png');

        if(e.x >= 1060+196 && e.x<=1296+196 && e.y<=687 && e.y>=590)

            this.botton\_quit = new Framework.Sprite(define.welcomimgPath + 'quit\_game\_touch.png');

        else

            this.botton\_quit = new Framework.Sprite(define.welcomimgPath + 'quit\_game\_untouch.png');

        if(e.x >= 1060+196 && e.x<=1296+196 && e.y<=799 && e.y>=700)

            this.botton\_team = new Framework.Sprite(define.welcomimgPath + 'team\_game\_touch.png');

        else

            this.botton\_team = new Framework.Sprite(define.welcomimgPath + 'team\_game\_untouch.png');

        if(e.x >= 1060+196 && e.x<=1296+196 && e.y<=578 && e.y>=479)

            this.botton\_tutorial = new Framework.Sprite(define.welcomimgPath + 'tutorial\_game\_touch.png');

        else

            this.botton\_tutorial = new Framework.Sprite(define.welcomimgPath + 'tutorial\_game\_untouch.png');

        this.botton\_start.position = {

            x: 1050+196,

            y: 350

        };

        this.botton\_tutorial.position = {

            x: 1050+196,

            y: 460

        };

        this.botton\_team.position = {

            x: 1050+196,

            y: 680

        };

        this.botton\_quit.position = {

            x: 1050+196,

            y: 570

        };

        this.draw(Framework.Game.\_context);

    }

});

1. myMenu\_gameteam.js

var MyMenu\_gameteam = Framework.exClass(Framework.GameMainMenu , {

    //初始化loadingProgress需要用到的圖片

    initializeProgressResource: function() {

        this.loading = new Framework.Sprite(define.imagePath + 'loading.jpg');

        this.loading.position = {x: Framework.Game.getCanvasWidth() / 2 , y: Framework.Game.getCanvasHeight() / 2};

    },

    //在initialize時會觸發的事件

    loadingProgress: function(ctx, requestInfo) {

        this.loading.draw(ctx);

        ctx.font ='90px Arial';

        ctx.textAlign = 'center';

        ctx.fillStyle = 'white';

        ctx.fillText(Math.round(requestInfo.percent) + '%' , ctx.canvas.width / 2 , ctx.canvas.height / 2 + 300);

    },

    load: function() {

        this.go\_back\_menu = new Framework.Sprite(define.welcomimgPath + 'go\_back\_menu.png');

        this.audio = new Framework.Audio({

            start\_game: {

                mp3: define.musicPath + '遊戲開場音樂.mp3',

            }, keyup:{

                mp3: define.musicPath + '按按鍵.mp3',

            }, song2:{

                mp3: define.musicPath + '刀劍神域op1《crossing field》鋼琴曲.mp3',

            }

        });

        this.audio.play({name: 'song2', loop: true});

    },

    initialize: function() {

        this.go\_back\_menu.position = {

            x: Framework.Game.getCanvasWidth() / 2,

            y: Framework.Game.getCanvasHeight() / 2

        };

        this.go\_back\_menu.scale = 4;

        this.rootScene.attach(this.go\_back\_menu);

        this.rectPosition = {

            x: Framework.Game.getCanvasWidth() / 2 - 130,

            y: 0

        };

    },

    update:function(){

        this.rootScene.update();

    },

    draw: function(parentCtx) {

        parentCtx.textAlign = 'center';

        parentCtx.fillStyle = 'black';

        parentCtx.fillText("遊戲專案團隊", 750, 100);

        parentCtx.fillText("106820003 電資三 潘建蒼", 750, 200);

        parentCtx.fillText("106820046 電資三 凃昱安", 750, 300);

        parentCtx.fillText("指導老師：陳偉凱", 750, 700);

        parentCtx.fillText("PS.偉凱好帥", 750, 800);

        this.rootScene.draw(parentCtx);

        this.go\_back\_menu.draw(parentCtx);

    },

    click:function(e){

        this.audio.play({name: 'keyup', loop: false});

        if(e.x<915 && e.x>682 && e.y<573 && e.y>285){

            this.audio.stop('song2');

            Framework.Game.goToLevel('menu');

        }

    }

});

1. myMenu\_tutorial.js

var MyMenu\_tutorial = Framework.exClass(Framework.GameMainMenu , {

    //初始化loadingProgress需要用到的圖片

    initializeProgressResource: function() {

        this.loading = new Framework.Sprite(define.imagePath + 'loading.jpg');

        this.loading.position = {x: Framework.Game.getCanvasWidth() / 2 , y: Framework.Game.getCanvasHeight() / 2};

    },

    //在initialize時會觸發的事件

    loadingProgress: function(ctx, requestInfo) {

        this.loading.draw(ctx);

        ctx.font ='90px Arial';

        ctx.textAlign = 'center';

        ctx.fillStyle = 'white';

        ctx.fillText(Math.round(requestInfo.percent) + '%' , ctx.canvas.width / 2 , ctx.canvas.height / 2 + 300);

    },

    load: function() {

        this.go\_back\_menu = new Framework.Sprite(define.welcomimgPath + 'go\_back\_menu.png');

        this.audio = new Framework.Audio({

            start\_game: {

                mp3: define.musicPath + '遊戲開場音樂.mp3',

            }, keyup:{

                mp3: define.musicPath + '按按鍵.mp3',

            }, song2:{

                mp3: define.musicPath + '刀劍神域op1《crossing field》鋼琴曲.mp3',

            }

        });

        this.audio.play({name: 'song2', loop: true});

    },

    initialize: function() {

        this.go\_back\_menu.position = {

            x: Framework.Game.getCanvasWidth() \*0.75,

            y: Framework.Game.getCanvasHeight() \*0.25

        };

        this.go\_back\_menu.scale = 4;

        this.rootScene.attach(this.go\_back\_menu);

        this.rectPosition = {

            x: Framework.Game.getCanvasWidth() / 2 - 130,

            y: 0

        };

    },

    draw: function(parentCtx) {

        parentCtx.font = '24pt Algerian';

        parentCtx.textAlign = 'left';

        parentCtx.fillStyle = 'black';

        parentCtx.fillText("撿取：對著物品按空白鍵", 60, 50);

        parentCtx.fillText("丟棄：滑鼠需移至背包欲丟棄物品上，按D鍵", 60, 90);

        parentCtx.fillText("對話：對著npc按空白鍵", 60, 130);

        parentCtx.fillText("普攻(無裝備武器)：按S", 60, 170);

        parentCtx.fillText("突刺(裝備長矛)：按S", 60, 210);

        parentCtx.fillText("射箭(裝備弓箭)：按S", 60, 250);

        parentCtx.fillText("傳送(裝備空間法杖)：按S", 60, 290);

        parentCtx.fillText("放魔法(裝備火法杖、冰法杖)：長按S，直到詠唱完畢", 60, 330);

        parentCtx.fillText("吃食物：點擊背包中的食物", 60, 370);

        parentCtx.fillText("裝裝備：點擊背包中的裝備", 60, 410);

        parentCtx.fillText("合成：點擊合成欄位的合成紐", 60, 450);

        parentCtx.fillText("釣魚：裝備釣竿，對著池塘按F鍵，當人物出現\'!\'時，按空白鍵", 60, 490);

        parentCtx.fillText("挖植物：裝備鏟子，對著植物按空白鍵，適用目標：花、草、樹、樹苗和漿果", 60, 530);

        parentCtx.fillText("種植物：點擊背包中的植物(須為挖掘過的植物)", 60, 570);

        parentCtx.fillText("砍樹：裝備斧頭，對著樹木按空白鍵", 60, 610);

        parentCtx.fillText("挖礦：裝備十字搞，對著岩石按空白鍵", 60, 650);

        parentCtx.fillText("查看角色狀態：按E鍵", 60, 690);

        parentCtx.fillText("---------------------------------------------密技一---------------------------------------------", 60, 730);

        parentCtx.fillText("無敵：按P鍵", 60, 770);

        parentCtx.fillText("死亡：按Q鍵", 60, 810);

        parentCtx.fillText("勝利：按W鍵", 60, 850);

        parentCtx.fillText("召喚蜜蜂：按R鍵", 60, 890);

        parentCtx.fillText("召喚豬豬：按T鍵", 400, 770);

        parentCtx.fillText("召喚牛牛：按Y鍵", 400, 810);

        parentCtx.fillText("召喚眼球：按U鍵", 400, 850);

        parentCtx.fillText("召喚蝙蝠：按I鍵", 400, 890);

        parentCtx.fillText("召喚BOSS：按O鍵", 740, 770);

        parentCtx.fillText("大祕寶：往左走", 740, 810);

        //Q:死亡結束  W:勝利結束 R:蜜蜂 T:豬 Y:牛 U:眼球 I:蝙蝠 O:boss

        this.rootScene.draw(parentCtx);

        this.go\_back\_menu.draw(parentCtx);

    },

    click:function(e){

        this.audio.play({name: 'keyup', loop: false});

        if(e.x<1329 && e.x>1078 && e.y<348 && e.y>83){

            this.audio.stop('song2');

            Framework.Game.goToLevel('menu');

        }

    },

});

1. normal\_attack.js

var Normal\_attack = function(player, monster, playerWalkDirection, playerPositionOnMap) {

    this.player = player;

    this.monster = monster;

    this.attackSuccess = false;

    this.visitor = new ReduceDurabilityVisitor();

    this.attack = function(){

        this.attackSuccess = false;

        for(var i = 0;i < this.monster.length;i++){

            if(this.monster[i].is\_start){

                if(playerPositionOnMap.x + playerWalkDirection.x == this.monster[i].mapPosition.x && playerPositionOnMap.y + playerWalkDirection.y == this.monster[i].mapPosition.y){

                    this.monster[i].health -= this.player.character\_descruption\_total\_point[2];

                    this.monster[i].getHurt();

                    this.attackSuccess = true;

                    break;

                }

            }

        }

        if(this.attackSuccess && this.player.equipmentBar.equipmentList[2] != null)

            this.player.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

    }

};

1. npc\_event.js

var Npc\_event = function(map) {

    this.map = map;

    this.dialog\_blue = new Framework.Sprite(define.materialPath + 'dialog\_blue.png');

    this.dialog\_blue.scale = 1.5;

    this.mission\_block = new Framework.AnimationSprite({url: define.npcPath + '任務框.png', col:7 , row:1 , loop:false , speed:1.5});

    this.mission\_block.position = {x: (13)\*64, y: (6)\*64};

    this.mission\_block.scale = 0.8;

    this.dialog\_blue.position = {x: 800, y:700};

    this.npc = [];

    this.npc1 =  new Framework.Sprite(define.npcPath + '小丑.png');

    this.npc1.scale =  1.5;

    this.npc1.position = {x: 800, y:350};

    this.npc.push(this.npc1);

    this.npc1 =  new Framework.Sprite(define.npcPath + '主角.jpg');

    this.npc1.scale =  1.5;

    this.npc1.position = {x: 800, y:350};

    this.npc.push(this.npc1);

    this.npc1 =  new Framework.Sprite(define.npcPath + '莉莉.png');

    this.npc1.scale =  1.3;

    this.npc1.position = {x: 800, y:450};

    this.npc.push(this.npc1);

    this.npc\_name = "";

    this.drama = "";

    this.taking\_is\_start = false;

    this.amount = -1;

    this.talk\_des = "";

    this.mission\_chain = [];

    this.description = {

        "任務一":{

            "開始":"小丑哥哥",

            "劇本":[

                {key:"主角", des:"請問一下，這裡是...", finish:false, loop:false},

                {key:"小丑哥哥", des:"吵屁喔，沒看到我正在分莓果嗎？"},

                {key:"小丑哥哥", des:"有什麼問題先幫我把壞掉的莓果挑出來，拿給旁邊第二排第二個的遺跡裡的的莉莉，他會跟你說要怎麼做。"},

                {key:"小丑哥哥", des:"希望你找得到那個遺跡在哪裡，哈哈"},

                {key:"小丑哥哥", des:"什麼！你連怎麼撿東西都不會！？"},

                {key:"小丑哥哥", des:"你對著物品按空白鍵就撿起來了，很簡單吧"},

                {key:"主角", des:"好像真的蠻簡單的"},

                {key:"小丑哥哥", des:"那還不快去"},

                {key:"小丑哥哥", des:"最後提醒一下，如果按下\'E\'鍵，可以查看角色屬性。而且在左方角色狀態有一條灰色的" +

                                    "那是飢餓值，如果他它0的話，生命就會開始下降，要特別小心。"},

                {key:"主角", des:"好喔"},

            ],

            "說了":false

        },

        "任務二":{

            "開始":"商人莉莉", //攻擊、

            "劇本":[

                {key:"主角", des:"姊姊", finish:false, loop:false},

                {key:"商人莉莉", des:"!!!!"},

                {key:"商人莉莉", des:"嘴巴真甜呢，女孩。"+

                                "你知道我今年已經兩百歲了嗎，哈哈"},

                {key:"主角", des:"!!!!!!!!!!!!!!!!!!!!!!!!!!!!"},

                {key:"商人莉莉", des:"看在你這麼可愛的份上，就跟你說這個世界的運作方式。"},

                {key:"商人莉莉", des:"別管什麼莓果了"},

                {key:"商人莉莉", des:"撿起來的東西會放進背包裡，如果想要丟棄的話，你要先將滑鼠移到物品上，按\'D\'，這樣就丟掉了。"},

                {key:"主角", des:"好麻煩的操作方式喔！"},

                {key:"商人莉莉", des:"再講講合成方式，左邊的欄位是合成欄，按一下就可以合成物資"},

                {key:"商人莉莉", des:"但要小心背包裡如果沒有足夠的材料，是沒有辦法合成物品的"},

                {key:"主角", des:"好喔"},

                {key:"主角", des:"那要怎麼裝備裝備呢？"},

                {key:"商人莉莉", des:"問得漂亮。"},

                {key:"商人莉莉", des:"用滑鼠點擊背包裡的裝備，就可以裝備囉。"},

                {key:"主角", des:"這麼簡單喔"},

                {key:"商人莉莉", des:"但知道這些是不夠的，再跟你說說種植物、砍樹和挖植物"},

                {key:"商人莉莉", des:"挖植物之前你需要一把鏟子、裝備上後，朝著植物按空白鍵，多按幾下你就可以把它挖起囉。"},

                {key:"商人莉莉", des:"砍樹和挖植物也是同樣的道理，裝備上斧頭和十字鎬，對著樹木和岩石按空白鍵有各自的效果。"},

                {key:"主角", des:"啊不就好棒棒"},

                {key:"商人莉莉", des:"別吵啦，再說說釣魚，你需要一把釣魚竿，裝備後對著湖面按\'F\'，等你覺得差不多就可以按空白收竿囉。"},

                {key:"商人莉莉", des:"湖裡的魚可好吃了，你一定要吃吃看。"},

                {key:"商人莉莉", des:"最後說到攻擊方式"},

                {key:"商人莉莉", des:"有些武器是需要詠唱的，像是魔杖類武器，按下\'S\'，就可以施放魔法"},

                {key:"商人莉莉", des:"其他武器也是，按下\'S\'，就可以有不同的攻擊效果，但就不需要詠唱。"},

                {key:"商人莉莉", des:"好啦，介紹就到這邊，你可以走了"},

                {key:"商人莉莉", des:"記得要贏得遊戲，不但要生存下來，還要打倒魔王喔。"},

                {key:"商人莉莉", des:"魔王就住在最右下角的岩漿那裡，快去吧。"},

            ],

            "說了":false

        },

        "主角":{

            "picture":[

                {key:0, picture:this.npc[1]},

            ],

            "drama0":[

            {loop:false, key:0, des:"(...)", finish:false},

            {key:0, des:"(...睜開眼)"},

            {key:0, des:"這裡是......"},

            {key:1, des:"剛剛我不是還在學校嗎？"},

            {key:2, des:"這地方怎麼看起來這麼奇怪"},

            {key:3, des:"難不成我被綁架了！！！！！！"},

            {key:4, des:"啊那邊有一個看起來很好心的人"},

            {key:5, des:"去問問他好了"},]

        },

        "小丑哥哥":{

            "picture":[

                {key:0, picture:this.npc[0]},

            ],

            "dialog":[

                {loop:true, des:"你知道", finish:false},

                {key:1, des:"偉凱的課很難修"},

                {key:2, des:"很多人都會當了"},

                {key:3, des:"你要不要放棄"},

            ]

        },

        "商人莉莉":{

            "picture":[

                {key:0, picture:this.npc[2]},

            ],

            "dialog":[

                {loop:true, des:"還不快去", finish:false},

                {key:1, des:"不打倒魔王的話"},

                {key:2, des:"就永遠逃離不了這裡了"},

                {key:3, des:"你知道嗎？"},]

        }

    };

    this.mission\_chain.push(this.description['任務一']);

    this.mission\_chain.push(this.description['任務二']);

    this.npc\_position = {x:0, y:0};

    this.characterHasMission = function(){

        switch(this.mission\_chain[0]['開始']){

            case "小丑哥哥":

                this.npc\_position = {x: 49, y:47}

                break;

            case "商人莉莉":

                this.npc\_position = {x: 20, y:15}

                break;

            default:

                break

        }

    }

    this.characterHasMission();

    this.checkMissionBlockHasStart = function(){

        if( Math.abs(this.npc\_position.x - map.playerPositionOnMap.x)<=5 &&  Math.abs(this.npc\_position.y - map.playerPositionOnMap.y)<=5){

            return true;;

        }else{

            return false;

        }

    }

    this.trigger = function(name, drama\_name){

        this.npc\_name = name;

        this.taking\_is\_start = true;

        if(this.mission\_chain[0]["開始"] == this.npc\_name && !this.mission\_chain[0]["說了"]){

            this.drama = -999;

        }else

            this.drama = drama\_name;

        this.amount = -1;

    }

    this.talking = function(){

        this.amount ++;

        if(this.drama == -999){

            if(this.mission\_chain[0]["劇本"][0].finish)

                this.taking\_is\_start = false;

            else{

                if(this.amount >=  this.mission\_chain[0]["劇本"].length){

                    this.taking\_is\_start = false;

                    this.amount = 0;

                    this.mission\_chain[0]["說了"] = true;

                    if(! this.mission\_chain[0]["劇本"][0].finish)

                        if(! this.mission\_chain[0]["劇本"][0].loop)

                            this.mission\_chain[0]["劇本"][0].finish = true;

                    else

                        this.taking\_is\_start = false;

                    if(this.mission\_chain.length > 1)

                        this.mission\_chain.splice(0, 1);

                    else

                        this.mission\_block.stop();

                    this.characterHasMission();

                }else

                    this.talk\_des = this.mission\_chain[0]["劇本"][this.amount].des;

            }

        }else{

            if(this.description[this.npc\_name][this.drama][0].finish)

                this.taking\_is\_start = false;

            else{

                if(this.amount >= this.returnSayLong()){

                    this.taking\_is\_start = false;

                    this.amount = 0;

                    if(!this.description[this.npc\_name][this.drama][0].finish){

                        if(!this.description[this.npc\_name][this.drama][0].loop)

                            this.description[this.npc\_name][this.drama][0].finish = true;

                    }else

                        this.taking\_is\_start = false;

                }else

                    this.talk\_des = this.description[this.npc\_name][this.drama][this.amount].des;

            }

        }

    }

    this.returnSayLong = function(){

        return this.description[this.npc\_name][this.drama].length;

    }

    this.update = function(){

        this.mission\_block.update();

    }

    if(this.mission\_chain.length != 0){

        this.mission\_block.start({ from: 0, to: 2, loop: true});

    }

    this.draw = function(ctx){

        if(this.checkMissionBlockHasStart()){

            this.mission\_block.position = {x:(13+this.npc\_position.x - this.map.playerPositionOnMap.x)\*64+16,y:(7+this.npc\_position.y - this.map.playerPositionOnMap.y)\*64-32};

            if(this.mission\_block.\_start)

                this.mission\_block.draw(ctx);

        }

        if(this.taking\_is\_start){

            ctx.textAlign = 'center';

            ctx.font = "40px Arial";

            ctx.fillStyle = "gold";

            if(this.drama == -999){

                this.description[ this.mission\_chain[0]["劇本"][this.amount].key ]["picture"][0].picture.draw(ctx);

                this.dialog\_blue.draw(ctx);

                ctx.fillText(this.mission\_chain[0]["劇本"][this.amount].key,370 ,615);

            }

            else{

                this.description[this.npc\_name]["picture"][0].picture.draw(ctx);

                this.dialog\_blue.draw(ctx);

                ctx.fillText(this.npc\_name,370 ,615);

            }

            ctx.font = "30px Arial";

            ctx.fillStyle = "white";

            for(var i=0,j=0,k=0;i<this.talk\_des.length;i++){

                if(i%35 == 0){

                    j +=35;

                    k=0;

                }

                ctx.fillText(this.talk\_des.charAt(i),290 +k\*30,635+j);

                k++;

            }

        }

    }

};

Object.defineProperty(Npc\_event.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.Npc\_event.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. npc1.js

var Npc1 = function(map) {

    this.npc1 = new Framework.AnimationSprite({url:define.materialPath + 'character1.png',

        col:3 , row:4 , loop:true , speed:12});

    this.npc1.scale = 1.5;

    this.npc1.index = 1;

    this.item\_num = -10;

    var PIXEL\_CONST = 64;

    this.name = "小丑哥哥";

    this.mapPosition = {x:0, y:0};

    this.walkTarget = {x:0, y:0};

    this.spritePosition = {x:0, y:0};

    this.constants = new Constants();

    this.map = map;

    this.isdieing = false;

    this.isdead = false;

    this.dieingCounter = 0;

    this.is\_start = false;

    this.canWalking = true;

    this.isWalking = false;

    this.health = 200;

    this.attack = 5;

    var m\_monster = this;

    this.monsterDirection = {x:0, y:0};

    this.walk = function(moveStep){

        if(this.isWalking === false){

            this.isWalking = true;

            this.walkTarget = {x:this.mapPosition.x + moveStep.x, y:this.mapPosition.y + moveStep.y};

            this.changeWalkDirection(moveStep);

        }

    }

    this.changeWalkDirection = function(walkStep){

        if(walkStep.x > 0){

            this.playerDirection = this.constants.DirectionEnum.RIGHT;

            this.monsterDirection = this.constants.Direction.RIGHT;

        }else if(walkStep.x <0){

            this.playerDirection = this.constants.DirectionEnum.LEFT;

            this.monsterDirection = this.constants.Direction.LEFT;

        }

        if(walkStep.y > 0){

            this.playerDirection = this.constants.DirectionEnum.DOWN;

            this.monsterDirection = this.constants.Direction.DOWN;

        }else if(walkStep.y < 0){

            this.playerDirection = this.constants.DirectionEnum.UP;

            this.monsterDirection = this.constants.Direction.UP;

        }

        this.npc1.start({ from: this.playerDirection \* 3, to: this.playerDirection \* 3 + 2, loop: true});

    }

    this.die = function(){

        this.isdead = true;

    }

    this.stopWalk = function()

    {

        this.canWalking = false;

    }

    this.walkEnd = function(){    }

    var walkSpeed = 8;

    this.walkAlittle = function(){

        if(this.playerDirection === this.constants.DirectionEnum.DOWN)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y + walkSpeed};

        else if(this.playerDirection === this.constants.DirectionEnum.LEFT)

            this.spritePosition = {x:this.spritePosition.x - walkSpeed, y:this.spritePosition.y};

        else if(this.playerDirection === this.constants.DirectionEnum.RIGHT)

            this.spritePosition = {x:this.spritePosition.x + walkSpeed, y:this.spritePosition.y};

        else if(this.playerDirection === this.constants.DirectionEnum.UP)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y - walkSpeed};

    }

    this.update = function(){

        if(this.isdead ){ return; }

        this.checkIsMonsterOutCanvus();

        if(!this.is\_start){ return; }

        this.npc1.update();

        if(this.isWalking){

            if(this.walkTarget.x \* PIXEL\_CONST === this.spritePosition.x && this.walkTarget.y \* PIXEL\_CONST === this.spritePosition.y){

                this.isWalking = false;

                this.npc1.stop();

                this.npc1.index = this.playerDirection \* 3 + 1;

                this.mapPosition = this.walkTarget;

            }else{

                this.walkAlittle();

            }

        }else{

            if(this.canWalking && this.is\_start){

                // this.randomWalk();

            }

        }

    }

    this.draw = function(ctx){

        if(this.isdead){ return; }

        if(!this.is\_start){ return; }

        var xx = 13\*64 + this.spritePosition.x - this.map.playerPositionOnMap.x\*64;

        var yy = 7\*64 + this.spritePosition.y - this.map.playerPositionOnMap.y\*64;

        this.npc1.position = {x: xx, y: yy};

        this.npc1.draw(ctx);

    }

    var walkDir = 0;

    this.checkIsMonsterOutCanvus = function(){

        if( Math.abs((this.mapPosition.x-this.map.playerPositionOnMap.x)) <6 &&　Math.abs((this.mapPosition.y-this.map.playerPositionOnMap.y)) <6)

            this.is\_start = true;

        else

            this.is\_start = false;

    }

    this.randomWalk = function()

    {

        var randNum = Framework.Game.\_currentLevel.cycleCount % 553;

        walkDir++;

        var walkStep = {x:0,y:0};

        if(randNum % 133 == 0)

            walkStep.y = 1

        else if(randNum % 157 == 0)

            walkStep.y = -1

        else{

            walkDir = 0;

            return;

        }

        if(this.map.checkMonsterIsWalkAble( {x: this.mapPosition.x + walkStep.x, y:this.mapPosition.y + walkStep.y} ))

            this.walk(walkStep);

    }

};

Object.defineProperty(Npc1.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.spritePosition = {x:this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

Object.defineProperty(Npc1.prototype, 'isDead', {

    get: function() {

        return this.isdead;

    }

});

1. npc2.js

var Npc2 = function(map) {

    this.npc1 = new Framework.AnimationSprite({url:define.materialPath + '莉莉.png',

        col:3 , row:4 , loop:true , speed:12});

    this.npc1.scale = 1.5;

    this.npc1.index = 1;

    this.item\_num = -11;

    var PIXEL\_CONST = 64;

    this.name = "商人莉莉";

    this.mapPosition = {x:0, y:0};

    this.walkTarget = {x:0, y:0};

    this.spritePosition = {x:0, y:0};

    this.constants = new Constants();

    this.map = map;

    this.isdieing = false;

    this.isdead = false;

    this.dieingCounter = 0;

    this.is\_start = false;

    this.canWalking = true;

    this.isWalking = false;

    this.health = 200;

    this.attack = 5;

    var m\_monster = this;

    this.monsterDirection = {x:0, y:0};

    this.walk = function(moveStep){

        if(this.isWalking === false){

            this.isWalking = true;

            this.walkTarget = {x:this.mapPosition.x + moveStep.x, y:this.mapPosition.y + moveStep.y};

            this.changeWalkDirection(moveStep);

        }

    }

    this.changeWalkDirection = function(walkStep){

        if(walkStep.x > 0){

            this.playerDirection = this.constants.DirectionEnum.RIGHT;

            this.monsterDirection = this.constants.Direction.RIGHT;

        }else if(walkStep.x <0){

            this.playerDirection = this.constants.DirectionEnum.LEFT;

            this.monsterDirection = this.constants.Direction.LEFT;

        }

        if(walkStep.y > 0){

            this.playerDirection = this.constants.DirectionEnum.DOWN;

            this.monsterDirection = this.constants.Direction.DOWN;

        }else if(walkStep.y < 0){

            this.playerDirection = this.constants.DirectionEnum.UP;

            this.monsterDirection = this.constants.Direction.UP;

        }

        this.npc1.start({ from: this.playerDirection \* 3, to: this.playerDirection \* 3 + 2, loop: true});

    }

    this.die = function(){

        this.isdead = true;

    }

    this.stopWalk = function(){

        this.canWalking = false;

    }

    var walkSpeed = 8;

    this.walkAlittle = function(){

        if(this.playerDirection === this.constants.DirectionEnum.DOWN)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y + walkSpeed};

        else if(this.playerDirection === this.constants.DirectionEnum.LEFT)

            this.spritePosition = {x:this.spritePosition.x - walkSpeed, y:this.spritePosition.y};

        else if(this.playerDirection === this.constants.DirectionEnum.RIGHT)

            this.spritePosition = {x:this.spritePosition.x + walkSpeed, y:this.spritePosition.y};

        else if(this.playerDirection === this.constants.DirectionEnum.UP)

            this.spritePosition = {x:this.spritePosition.x, y:this.spritePosition.y - walkSpeed};

    }

    this.update = function(){

        if(this.isdead ){ return; }

        this.checkIsMonsterOutCanvus();

        if(!this.is\_start){ return; }

        this.npc1.update();

        if(this.isWalking){

            if(this.walkTarget.x \* PIXEL\_CONST === this.spritePosition.x && this.walkTarget.y \* PIXEL\_CONST === this.spritePosition.y){

                this.isWalking = false;

                this.npc1.stop();

                this.npc1.index = this.playerDirection \* 3 + 1;

                this.mapPosition = this.walkTarget;

            }else

                this.walkAlittle();

        }else{

            if(this.canWalking && this.is\_start){

                // this.randomWalk();

            }

        }

    }

    this.draw = function(ctx){

        if(this.isdead){ return; }

        if(!this.is\_start){ return; }

        var xx = 13\*64 + this.spritePosition.x - this.map.playerPositionOnMap.x\*64;

        var yy = 7\*64 + this.spritePosition.y - this.map.playerPositionOnMap.y\*64;

        this.npc1.position = {x: xx, y: yy};

        this.npc1.draw(ctx);

    }

    var walkDir = 0;

    this.checkIsMonsterOutCanvus = function(){

        if( Math.abs((this.mapPosition.x-this.map.playerPositionOnMap.x)) <6 &&　Math.abs((this.mapPosition.y-this.map.playerPositionOnMap.y)) <6)

            this.is\_start = true;

        else

            this.is\_start = false;

    }

    this.randomWalk = function()

    {   var randNum = Framework.Game.\_currentLevel.cycleCount % 553;

        walkDir++;

        var walkStep = {x:0,y:0};

        if(randNum % 133 == 0)

            walkStep.y = 1

        else if(randNum % 157 == 0)

            walkStep.y = -1

        else{

            walkDir = 0;

            return;

        }

        if(this.map.checkMonsterIsWalkAble( {x: this.mapPosition.x + walkStep.x, y:this.mapPosition.y + walkStep.y} ))

            this.walk(walkStep);

    }

};

Object.defineProperty(Npc2.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.spritePosition = {x:this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

Object.defineProperty(Npc2.prototype, 'isDead', {

    get: function() {

        return this.isdead;

    }

});

1. null\_attack.js

var Null\_attack = function() {

    this.attackSuccess = false;

    this.attack = function(){

        console.log("Attack didn't success.");

    }

};

1. object\_factory.js

var Object\_factory = function() {};

Object\_factory.prototype.objectClass = null;

Object\_factory.prototype.createObject = function(options){

  if(options.type == "Tool")

    this.objectClass = Tool;

  else if(options.type == "Weapon")

    this.objectClass = Weapon;

  else

    return;

  return new this.objectClass(options);

};

Object\_factory.prototype.createMonster = function(map){

  var random = Math.floor(Math.random()\*5);

  switch(random){

    case 0:

      this.objectClass = Monster\_cute\_little\_eye;

      break;

    case 1:

      this.objectClass = Monster\_pig;

      break;

    case 2:

      this.objectClass = Monster\_cow;

      break;

    case 3:

      this.objectClass = Monster\_bee;

      break;

    case 4:

      this.objectClass = Monster\_bat;

      break;

    default:

      return;

  }

  return new this.objectClass(map);

};

1. player\_hide\_animation.js

var Player\_hide\_animation = function() {

    this.url = define.skillAnimationPath + 'Recovery1.png';

    this.hideAnimation = new Framework.AnimationSprite({url:this.url, col:5 , row:6 , loop:false , speed:12});

    this.hideAnimation.scale = 1;

    this.hideAnimation.position = {x:11\*64+32, y:5\*64-24};

    this.hideAnimation.index = 0;

    this.update = function(){

        this.hideAnimation.update();

    }

    this.draw = function(ctx){

        this.hideAnimation.draw(ctx);

    }

    this.start = function(){

        this.hideAnimation.start({ from: 0, to: 29, loop: false});

    }

};

1. proxy.js

var Proxy = function() {

    this.itemArray ;

    this.map\_name ;

    this.mapArray = [

        //1

        [ [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,91,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,255,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [255,200,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,255,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [255,200,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

        ],

        //2

        [ [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [200,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,196,196,196,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,196,196,196,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,196,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,91,91,91,91,91,91],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,200,200,200,200,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255],

            [91,91,91,91,91,91,91,91,91,91,196,196,196,196,196,196,196,196,196,196,91,91,91,91,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,196],

        ],

.

.

.

//100

        [ [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        ],

    ];

    this.whichToDetect = function(map\_name, itemArray){

        this.itemArray = itemArray

        this.map\_name = map\_name;

    }

    this.objectFactory = new Object\_factory();

    this.detect = function(map\_number){

        if(this.itemArray[map\_number] == null){

            this.createItemMap(map\_number);

            this.collectNullMap(map\_number);

        }

    }

    this.null\_map = [];

    this.nullMapClean = function(){

        this.null\_map = [];

    }

    this.collectNullMap = function(map\_number){

        this.null\_map.push(map\_number);

    }

    this.item;

    //1:小花 2:蜘蛛網 3:石頭 4:樹枝 5:偉凱的作業簿 6:草 7:木頭 8:燧石 9:黃金 10:豬皮

    //11:蜂刺 12:雪球 13:繩索 14:露水 15:十字鎬 16:斧頭 17:釣魚竿 18:鏟子 19:黃金斧頭 20:黃金鏟子

    //21:黃金十字鎬 22:頭盔 23:草製盔甲 24:木製盔甲 25:長矛 26:吹箭 27:國王法杖 28:空間法杖 29:火法杖 30:冰法杖

    //31:黃金提燈 32:火把 33:帳篷 34:篝火 35:冰塊 36:漿果叢 37:採摘的花 38:採摘的草 39:漿果 40:挖的漿果叢(拔)

    //41:挖的花(拔) 42:挖的樹 43:挖的草(拔) 44:挖的漿果叢(未拔) 45:挖的花(未拔) 46:挖的草(未拔) 47:挖的樹苗(拔)

    //48:挖的樹苗(未拔) 49:蝙蝠翅膀 50:眼球 51:蜂蜜 52:大肉 53:怪物肉 54:黑漿果

    //-1:樹 -3:假空白 -4:小樹苗 -10:小丑哥哥 -11:商人莉莉

    this.randomCreateItem = function(i,j,k){

        switch(this.mapArray[i][j][k]){

            case 91:

                return new Item\_blank();

            case 196:

                return new Item\_blank();

            case 200:

                return new Item\_blank();

            case 192:

                var random = Math.floor(Math.random()\*150);

                if(random == 0 || random == 10 || random == 20)

                    return new Item\_sapling();

                else if(random == 11 || random == 12)

                    return new Item\_grass();

                else if(random == 50)

                    return new Item\_bush();

                else if(random == 60)

                    return new Item\_flint();

                else if(random == 70)

                    return new Item\_flower();

                else

                    return new Item\_blank();

            case 123:

                var random = Math.floor(Math.random()\*150);

                if(random == 0 || random == 10 || random == 20 || random == 30)

                    return new Map\_item\_tree();

                else if(random == 11)

                    return new Item\_grass();

                else if(random == 50)

                    return new Item\_bush();

                else if(random == 60)

                    return new Item\_flint();

                else if(random == 70)

                    return new Item\_sapling();

                else

                    return new Item\_blank();

            case 137:

                var random = Math.floor(Math.random()\*100);

                if(random == 60)

                    return new Item\_flint();

                else

                    return new Item\_blank();

            case 255:

                var random = Math.floor(Math.random()\*200);

                if(random == 10)

                    return new Item\_flint();

                else if(random == 11)

                    return new Item\_grass();

                else if(random == 50)

                    return new Item\_bush();

                else if(random == 60 || random == 90)

                    return new Item\_snow\_ball();

                else if(random == 70)

                    return new Item\_sapling();

                else

                    return new Item\_blank();

            default:

                return new Item\_blank();

        }

    }

    this.createItemMap= function(map\_number){

        var i = map\_number;

        this.itemArray\_two\_dimension = [];

        this.itemArray\_one\_dimension = [];

        switch(this.map\_name){

            case "World":

                this.item = this.item\_map\_Array;

                break;

            case "House1":

                this.item = this.house\_item\_map\_Array;

                break;

            default:

                break;

        }

        for(var j=0;j<this.item[i].length;j++){

            for(var k=0;k<this.item[i][j].length;k++){

                switch(this.item[i][j][k]){

                    case 0:

                        if(this.map\_name == "World"){

                            this.itemArray\_one\_dimension.push(this.randomCreateItem(i,j,k));

                            break;

                        }else{

                            this.itemArray\_one\_dimension.push(new Item\_blank());

                            break;

                        }

                    case 1:

                        this.itemArray\_one\_dimension.push(new Item\_flower());

                        break;

                    case 2:

                        this.itemArray\_one\_dimension.push(new Item\_spider\_web());

                        break;

                    case 3:

                        this.itemArray\_one\_dimension.push(new Item\_stone());

                        break;

                    case 4:

                        this.itemArray\_one\_dimension.push(new Item\_branch());

                        break;

                    case 5:

                        this.itemArray\_one\_dimension.push(new Item\_waikei\_homework());

                        break;

                    case 6:

                        this.itemArray\_one\_dimension.push(new Item\_grass());

                        break;

                    case 7:

                        this.itemArray\_one\_dimension.push(new Item\_wood());

                        break;

                    case 8:

                        this.itemArray\_one\_dimension.push(new Item\_flint());

                        break;

                    case 9:

                        this.itemArray\_one\_dimension.push(new Item\_gold());

                        break;

                    case 10:

                        this.itemArray\_one\_dimension.push(new Item\_pigskin());

                        break;

                    case 11:

                        this.itemArray\_one\_dimension.push(new Item\_bee\_sting());

                        break;

                    case 12:

                        this.itemArray\_one\_dimension.push(new Item\_snow\_ball());

                        break;

                    case 13:

                        this.itemArray\_one\_dimension.push(new Item\_rope());

                        break;

                    case 14:

                        this.itemArray\_one\_dimension.push(new Item\_droplet());

                        break;

                    case 15:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_pixilart.png'),

                                attackPoint: 1,

                                itemNumber: 15,

                                reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                            }

                        ));

                        break;

                    case 16:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_ax.png'),

                                attackPoint: 1,

                                itemNumber: 16,

                                reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                            }

                        ));

                        break;

                    case 17:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_fishing\_rod.png'),

                                attackPoint: 1,

                                itemNumber: 17,

                                reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                            }

                        ));

                        break;

                    case 18:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_shovel.png'),

                                attackPoint: 1,

                                itemNumber: 18,

                                reduceDurability: function(visitor){visitor.visitBasicTool(this)}

                            }

                        ));

                        break;

                    case 19:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_ax.png'),

                                attackPoint: 2,

                                itemNumber: 19,

                                reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

                            }

                        ));

                        break;

                    case 20:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_shovel.png'),

                                attackPoint: 2,

                                itemNumber: 20,

                                reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

                            }

                        ));

                        break;

                    case 21:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Tool",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_gold\_pixilart.png'),

                                attackPoint: 2,

                                itemNumber: 21,

                                reduceDurability: function(visitor){visitor.visitGoldenTool(this)}

                            }

                        ));

                        break;

                    case 22:

                        this.itemArray\_one\_dimension.push(new Item\_helmat());

                        break;

                    case 23:

                        this.itemArray\_one\_dimension.push(new Item\_armor());

                        break;

                    case 24:

                        this.itemArray\_one\_dimension.push(new Item\_wood\_armor());

                        break;

                    case 25:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Weapon",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_spear.png'),

                                attack: [3,0,0],

                                itemNumber: 25,

                                scale: 0.8,

                                reduceDurability: function(visitor){visitor.visitSpear(this)}

                            }

                        ));

                        break;

                    case 26:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Weapon",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_arror.png'),

                                attack: [0,0,3],

                                itemNumber: 26,

                                scale: 2,

                                reduceDurability: function(visitor){visitor.visitArror(this)}

                            }

                        ));

                        break;

                    case 27:

                        this.itemArray\_one\_dimension.push(new Item\_king\_wand());

                        break;

                    case 28:

                        this.itemArray\_one\_dimension.push(new Item\_space\_wand());

                        break;

                    case 29:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Weapon",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_fire\_wand.png'),

                                attack: [0,2,0],

                                itemNumber: 29,

                                scale: 0.8,

                                reduceDurability: function(visitor){visitor.visitWand(this)}

                            }

                        ));

                        break;

                    case 30:

                        this.itemArray\_one\_dimension.push(this.objectFactory.createObject(

                            {

                                type: "Weapon",

                                imagePath: new Framework.Sprite(define.materialPath + 'item\_ice\_wand.png'),

                                attack: [0,2,0],

                                itemNumber: 30,

                                scale: 0.8,

                                reduceDurability: function(visitor){visitor.visitWand(this)}

                            }

                        ));

                        break;

                    case 31:

                        this.itemArray\_one\_dimension.push(new Item\_lamp());

                        break;

                    case 32:

                        this.itemArray\_one\_dimension.push(new Item\_firebundle());

                        break;

                    case 33:

                        this.itemArray\_one\_dimension.push(new Item\_camp());

                        break;

                    case 34:

                        this.itemArray\_one\_dimension.push(new Item\_campfire());

                        break;

                    case 35:

                        this.itemArray\_one\_dimension.push(new Item\_ice());

                        break;

                    case 36:

                        this.itemArray\_one\_dimension.push(new Item\_bush());

                        break;

                    case 37:

                        this.itemArray\_one\_dimension.push(new Item\_flower\_picked());

                        break;

                    case 38:

                        this.itemArray\_one\_dimension.push(new Item\_bush());

                        break;

                    case 39:

                        this.itemArray\_one\_dimension.push(new Item\_berry());

                        break;

                    case 54:

                        this.itemArray\_one\_dimension.push(new Item\_black\_berry());

                        break;

                    case -1:

                        this.itemArray\_one\_dimension.push(new Map\_item\_tree(this));

                        break;

                    case -3:

                        this.itemArray\_one\_dimension.push(new Item\_fake());

                        break;

                    case -4:

                        this.itemArray\_one\_dimension.push(new Item\_sapling());

                        break;

                    case -5:

                        this.itemArray\_one\_dimension.push(new Building\_remance(0));

                        break;

                    case -6:

                        this.itemArray\_one\_dimension.push(new Building\_remance(1));

                        break;

                    case -7:

                        this.itemArray\_one\_dimension.push(new Building\_remance(2));

                        break;

                    default:

                        this.itemArray\_one\_dimension.push(new Item\_blank());

                        break;

                }

            }

            this.itemArray\_two\_dimension.push(this.itemArray\_one\_dimension);

            this.itemArray\_one\_dimension = [];

        }

        this.itemArray[map\_number] = this.itemArray\_two\_dimension;

    }

};

1. reduceDurabilityVisitor.js

var ReduceDurabilityVisitor = function() {

    this.visitBasicTool = function(basicTool){

        basicTool.durability -= 10;

    }

    this.visitGoldenTool = function(goldenTool){

        goldenTool.durability -= 5;

    }

    this.visitFireBundle = function(fireBundle){

        if(fireBundle.inEquipmentbar){

            fireBundle.durability -= 1;

            setTimeout(()=>{

                fireBundle.reduceDurability(this);

            },1000);

        }

    }

    this.visitArror = function(arror){

        arror.durability -= 10;

    }

    this.visitSpear = function(arror){

        arror.durability -= 5;

    }

    this.visitWand = function(wand){

        wand.durability -= 20;

    }

};

1. score.js

var Score = function() {

    this.score = 500;

    this.scoreToDraw = 0;

    this.position = {

        x: Framework.Game.getCanvasWidth() / 2,

        y: Framework.Game.getCanvasHeight() / 2

    };;

    this.background = new Framework.Sprite(define.materialPath + 'DecorativeTile.png');

    this.background.scale = 2;

    this.background.position = this.position;

    this.frame = new Framework.Sprite(define.materialPath + 'Floral2.png');

    this.frame.scale = 2;

    this.frame.position = this.position;

    this.kill = 0;

    this.synthesis = 0;

    this.state = 0;

    this.scoreAddBySynthesis = function(){

        this.score += 2;

        this.synthesis += 1;

    }

    this.scoreAddByKillMonster = function(){

        this.score += 4;

        this.kill += 1;

    }

    this.drawScore = function(){

        var interval = setInterval(()=>{

            if(this.scoreToDraw <= this.score){

                this.draw(Framework.Game.\_context);

                this.scoreToDraw += 2;

                if(this.scoreToDraw > this.score)

                    clearInterval(interval);

            }

        },10);

        var stateInterval = setInterval(()=>{

            this.state += 1;

            if(this.state == 3)

                clearInterval(stateInterval);

        },350);

    }

    this.draw = function(ctx){

        this.background.draw(ctx);

        this.frame.draw(ctx);

        ctx.font = '90pt Algerian';

        ctx.fillStyle = 'yellow';

        ctx.strokeStyle = 'blue';

        ctx.lineWidth = 2.5;

        if(this.state == 0){

            ctx.fillText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y);

            ctx.strokeText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y);

        }else if(this.state == 1){

            ctx.fillText("Kill: " + this.kill, this.position.x, this.position.y+64);

            ctx.strokeText("Kill: " + this.kill, this.position.x, this.position.y+64);

            ctx.fillText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y-64);

            ctx.strokeText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y-64);

        }else if(this.state == 2){

            ctx.fillText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y-128);

            ctx.strokeText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y-128);

            ctx.fillText("Kill: " + this.kill, this.position.x, this.position.y);

            ctx.strokeText("Kill: " + this.kill, this.position.x, this.position.y);

            ctx.fillText("Synthesis: " + this.synthesis, this.position.x, this.position.y+128);

            ctx.strokeText("Synthesis: " + this.synthesis, this.position.x, this.position.y+128);

        }else{

            ctx.fillText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y-196);

            ctx.strokeText("Day: " + Framework.Game.\_levels[2].level.map.clock.day, this.position.x, this.position.y-196);

            ctx.fillText("Kill: " + this.kill, this.position.x, this.position.y-64);

            ctx.strokeText("Kill: " + this.kill, this.position.x, this.position.y-64);

            ctx.fillText("Synthesis: " + this.synthesis, this.position.x, this.position.y+64);

            ctx.strokeText("Synthesis: " + this.synthesis, this.position.x, this.position.y+64);

            ctx.fillText("Score: " + this.scoreToDraw, this.position.x, this.position.y+196);

            ctx.strokeText("Score: " + this.scoreToDraw, this.position.x, this.position.y+196);

        }

    }

};

1. skill\_handler.js

var Skill\_handler = function() {

    this.url = define.skillAnimationPath + 'fire\_wand\_level1.png';

    this.fire\_wand\_level1 = new Framework.AnimationSprite({url:this.url, col:5 , row:3 , loop:false , speed:12});

    this.fire\_wand\_level1.scale = 1.25;

    this.fire\_wand\_level1.index = 0;

    this.fire\_wand\_level1.position = {x:13\*64-64\*3, y:7\*64};

    this.url2 = define.skillAnimationPath + 'Ice1.png';

    this.ice\_wand\_level1 = new Framework.AnimationSprite({url:this.url2, col:5 , row:3 , loop:false , speed:12});

    this.ice\_wand\_level1.scale = 1.25;

    this.ice\_wand\_level1.index = 0;

    this.ice\_wand\_level1.position = {x:13\*64-64\*3, y:7\*64};

    this.update = function(){

        this.fire\_wand\_level1.update();

        this.ice\_wand\_level1.update();

    }

    this.isAnimationStart = function(){

        return this.fire\_wand\_level1.\_start || this.ice\_wand\_level1.\_start

    }

    this.start = function(playerWalkDirection,playerPositionOnMap,equipmentId){

        var audio = new Framework.Audio({

            fire: {

                mp3: define.musicPath + '火魔法.mp3',

            },

            ice: {

                mp3: define.musicPath + '冰魔法.mp3',

            }

        });

        this.mapPosition = {x: playerPositionOnMap.x+playerWalkDirection.x\*3, y: playerPositionOnMap.y+playerWalkDirection.y\*3};

        if(equipmentId == 29){

            this.fire\_wand\_level1.start({ from: 0, to: 14, loop: false});

        }

        if(equipmentId == 30){

            this.ice\_wand\_level1.start({ from: 0, to: 14, loop: false});

        }

    }

    this.setPosition = function(positionOnMap){

        this.fire\_wand\_level1.position = positionOnMap;

        this.ice\_wand\_level1.position = positionOnMap;

    }

    this.draw = function(ctx){

        if(this.fire\_wand\_level1.\_start)

            this.fire\_wand\_level1.draw(ctx);

        if(this.ice\_wand\_level1.\_start)

            this.ice\_wand\_level1.draw(ctx);

    }

};

1. skill\_timer.js

var Skill\_timer = function() {

    this.magic\_time\_counter = new Framework.Sprite(define.skillAnimationPath + 'magic\_time\_counter.png');

    this.magic\_time\_counter.scale = 2;

    this.magic\_time\_counter.position = {x: 13\*64, y: 7\*64-32};

    this.maxMagicEnergy = 175;

    this.currentMagicEnergy = 0;

    this.buttonPress = false;

    this.isEnergyFull = false;

    this.startAccumulateEnergy = function(){

        this.buttonPress = true;

        var interval = setInterval(()=>{

            if(this.buttonPress){

                if(this.currentMagicEnergy < 175)

                    this.currentMagicEnergy += 2.5;

                if(this.currentMagicEnergy >= 175)

                    this.isEnergyFull = true;

            }

            else

                clearInterval(interval);

        },50);

    }

    this.stopAccumulateEnergy = function(){

        this.buttonPress = false;

        this.isEnergyFull = false;

        this.currentMagicEnergy = 0;

    }

    this.clear = function(){

        this.buttonPress = false;

        this.currentMagicEnergy = false;

        this.isEnergyFull = false;

    }

    this.draw = function(ctx){

        ctx.beginPath();

        ctx.rect(746, 400, this.currentMagicEnergy, 17);

        ctx.fillStyle = "orange";

        ctx.fill();

        this.magic\_time\_counter.draw(ctx);

    }

};

1. spear\_handler.js

var Spear\_handler = function() {

    this.url = define.skillAnimationPath + 'Cure4.png';

    this.spear = new Framework.AnimationSprite({url:this.url, col:5 , row:3 , loop:false , speed:18});

    this.spear.scale = 0.5;

    this.spear.index = 0;

    this.mapPosition;

    this.spear.position = {x:0, y:0};

    this.init = function(){

    }

    this.update = function(){

        this.spear.update();

    }

    this.start = function(playerWalkDirection,playerPositionOnMap){

        this.mapPosition = {x: playerPositionOnMap.x+playerWalkDirection.x, y: playerPositionOnMap.y+playerWalkDirection.y};

        this.spear.start({ from: 0, to: 3, loop: false});

    }

    this.draw = function(ctx){

        this.spear.draw(ctx);

    }

};

1. synthesisBar.js

var SynthesisBar = function(backpackList, score) {

    this.audio = new Framework.Audio({

         click\_change:{

            mp3: define.musicPath + '合成.mp3',

        }

    });

    this.backpack = new Framework.Sprite(define.materialPath + 'backpack.png');

    this.backpack.scale = 2;

    this.backpack.position = {x:0,y:0};

    this.currentPoint = {x:-1,y:-1};

    this.mousePosition;

    this.secondColumnRange = [];

    this.thirdColumnRange = [];

    this.firstColumeIndex = -1;

    this.secondColumnIndex = -1;

    this.item\_branch = new Framework.Sprite(define.materialPath + 'item\_branch.png');

    this.item\_branch.scale = 0.8;

    this.item\_bush\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_bush\_growed\_dig.png');

    this.item\_bush\_growed\_dig.scale = 2;

    this.item\_droplet = new Framework.Sprite(define.materialPath + 'item\_droplet.png');

    this.item\_droplet.scale = 2;

    this.item\_pixilart = new Framework.Sprite(define.materialPath + 'item\_pixilart.png');

    this.item\_pixilart.scale = 0.8;

    this.item\_gold\_pixilart = new Framework.Sprite(define.materialPath + 'item\_gold\_pixilart.png');

    this.item\_gold\_pixilart.scale = 0.8;

    this.item\_space\_wand = new Framework.Sprite(define.materialPath + 'item\_space\_wand.png');

    this.item\_space\_wand.scale = 0.8;

    this.item\_fire\_wand = new Framework.Sprite(define.materialPath + 'item\_fire\_wand.png');

    this.item\_fire\_wand.scale = 0.8;

    this.item\_king\_wand = new Framework.Sprite(define.materialPath + 'item\_king\_wand.png');

    this.item\_king\_wand.scale = 0.8;

    this.item\_ice\_wand = new Framework.Sprite(define.materialPath + 'item\_ice\_wand.png');

    this.item\_ice\_wand.scale = 0.8;

    this.item\_gold = new Framework.Sprite(define.materialPath + 'item\_gold.png');

    this.item\_gold.scale = 0.7;

    this.item\_spear = new Framework.Sprite(define.materialPath + 'item\_spear.png');

    this.item\_spear.scale = 0.8;

    this.item\_armor = new Framework.Sprite(define.materialPath + 'item\_armor.png');

    this.item\_armor.scale = 0.8;

    this.item\_wood\_armor = new Framework.Sprite(define.materialPath + 'item\_wood\_armor.png');

    this.item\_wood\_armor.scale = 0.8;

    this.item\_helmat = new Framework.Sprite(define.materialPath + 'item\_helmat.png');

    this.item\_helmat.scale = 0.8;

    this.item\_bee\_sting = new Framework.Sprite(define.materialPath + 'item\_bee\_sting.png');

    this.item\_bee\_sting.scale = 0.8;

    this.item\_ax = new Framework.Sprite(define.materialPath + 'item\_ax.png');

    this.item\_ax.scale = 0.8;

    this.item\_gold\_shovel = new Framework.Sprite(define.materialPath + 'item\_gold\_shovel.png');

    this.item\_gold\_shovel.scale = 0.8;

    this.item\_flint = new Framework.Sprite(define.materialPath + 'item\_flint.png');

    this.item\_flint.scale = 0.8;

    this.item\_snow\_ball = new Framework.Sprite(define.materialPath + 'item\_snow\_ball.png');

    this.item\_snow\_ball.scale = 2;

    this.item\_shovel = new Framework.Sprite(define.materialPath + 'item\_shovel.png');

    this.item\_shovel.scale = 0.8;

    this.item\_gold\_ax = new Framework.Sprite(define.materialPath + 'item\_gold\_ax.png');

    this.item\_gold\_ax.scale = 0.8;

    this.item\_pigskin = new Framework.Sprite(define.materialPath + 'item\_pigskin.png');

    this.item\_pigskin.scale = 0.8;

    this.item\_fishing\_rod = new Framework.Sprite(define.materialPath + 'item\_fishing\_rod.png');

    this.item\_fishing\_rod.scale = 0.8;

    this.item\_rope = new Framework.Sprite(define.materialPath + 'item\_rope.png');

    this.item\_rope.scale = 0.8;

    this.item\_camp = new Framework.Sprite(define.materialPath + 'item\_camp.png');

    this.item\_camp.scale = 0.8;

    this.item\_campfire = new Framework.Sprite(define.materialPath + 'item\_campfire.png');

    this.item\_campfire.scale = 2;

    this.item\_wood = new Framework.Sprite(define.materialPath + 'item\_wood.png');

    this.item\_wood.scale = 0.8;

    this.item\_lamp = new Framework.Sprite(define.materialPath + 'item\_lamp.png');

    this.item\_lamp.scale = 2;

    this.item\_bush = new Framework.Sprite(define.materialPath + 'item\_bush.png');

    this.item\_bush.scale = 2;

    this.item\_arror = new Framework.Sprite(define.materialPath + 'item\_arror.png');

    this.item\_arror.scale = 2;

    this.item\_ice = new Framework.Sprite(define.materialPath + 'item\_ice.png');

    this.item\_ice.scale = 0.8;

    this.item\_firebundle = new Framework.Sprite(define.materialPath + 'item\_firebundle.png');

    this.item\_firebundle.scale = 1.5;

    this.item\_spider\_web = new Framework.Sprite(define.materialPath + 'item\_spider\_web.png');

    this.item\_spider\_web.scale = 0.8;

    this.item\_grass = new Framework.Sprite(define.materialPath + 'item\_grass.png');

    this.item\_grass.scale = 2;

    this.item\_grass\_mowed = new Framework.Sprite(define.materialPath + 'item\_grass\_mowed.png');

    this.item\_grass\_mowed.scale = 2;

    this.item\_flower = new Framework.Sprite(define.materialPath + 'item\_flower.png');

    this.item\_flower.scale = 2;

    this.item\_flower\_pulled = new Framework.Sprite(define.materialPath + 'item\_flower\_pulled.png');

    this.item\_flower\_pulled.scale = 2;

    this.item\_stone = new Framework.Sprite(define.materialPath + 'item\_stone.png');

    this.item\_stone.scale = 2;

    this.item\_waikei\_homework = new Framework.Sprite(define.materialPath + 'item\_waikei\_homework.png');

    this.item\_waikei\_homework.scale = 2;

    this.item\_flower\_picked = new Framework.Sprite(define.materialPath + 'item\_flower.png');

    this.item\_flower\_picked.scale = 2;

    this.item\_grass\_picked = new Framework.Sprite(define.materialPath + 'item\_grass\_picked.png');

    this.item\_grass\_picked.scale = 0.8;

    //1:小花 2:蜘蛛網 3:石頭 4:樹枝 5:偉凱的作業簿 6:草 7:木頭 8:燧石 9:黃金 10:豬皮

    //11:蜂刺 12:雪球 13:繩索 14:露水 15:十字鎬 16:斧頭 17:釣魚竿 18:鏟子 19:黃金斧頭 20:黃金鏟子

    //21:黃金十字鎬 22:頭盔 23:草製盔甲 24:木製盔甲 25:長矛 26:吹箭 27:國王法杖 28:空間法杖 29:火法杖 30:冰法杖

    //31:黃金提燈 32:火把 33:帳篷 34:篝火 35:冰塊 36:漿果叢

    this.synthesisBar = [

        {item:"工具", position:{x:1, y:4}},

        {item:"戰鬥", position:{x:1, y:5}},

        {item:"技能", position:{x:1, y:6}},

        {item:"生存", position:{x:1, y:7}},

        {item:"精煉", position:{x:1, y:8}}

    ];

    this.synthesisBarDetail = [

        [

            {item: 15, position: {x:2,y:1}},

            {item: 16, position: {x:2,y:2}},

            {item: 17, position: {x:2,y:3}},

            {item: 18, position: {x:2,y:4}},

            {item: 19, position: {x:2,y:5}},

            {item: 20, position: {x:2,y:6}},

            {item: 21, position: {x:2,y:7}}

        ],

        [

            {item: 22, position: {x:2,y:3}},

            {item: 23, position: {x:2,y:4}},

            {item: 24, position: {x:2,y:5}},

            {item: 25, position: {x:2,y:6}},

            {item: 26, position: {x:2,y:7}}

        ],

        [

            {item: 27, position: {x:2,y:5}},

            {item: 28, position: {x:2,y:6}},

            {item: 29, position: {x:2,y:7}},

            {item: 30, position: {x:2,y:8}}

        ],

        [

            {item: 31, position: {x:2,y:6}},

            {item: 32, position: {x:2,y:7}},

            {item: 33, position: {x:2,y:8}},

            {item: 34, position: {x:2,y:9}}

        ],

        [

            {item: 13, position: {x:2,y:8}},

            {item: 35, position: {x:2,y:9}}

        ]

    ];

    this.synthesisBarMaterial = [

        [

            [{item: 4,amount: 2,position:{x:3, y:1}},{item: 8,amount: 2,position:{x:3, y:2}},{item:"合成",position:{x:3,y:3}}],

            [{item: 4,amount: 1,position:{x:3, y:2}},{item: 8,amount: 1,position:{x:3, y:3}},{item:"合成",position:{x:3,y:4}}],

            [{item: 4,amount: 2,position:{x:3, y:3}},{item: 2,amount: 2,position:{x:3, y:4}},{item:"合成",position:{x:3,y:5}}],

            [{item: 4,amount: 2,position:{x:3, y:4}},{item: 8,amount: 2,position:{x:3, y:5}},{item:"合成",position:{x:3,y:6}}],

            [{item: 4,amount: 1,position:{x:3, y:5}},{item: 8,amount: 1,position:{x:3, y:6}},{item: 9,amount: 1,position:{x:3, y:7}},{item:"合成",position:{x:3,y:8}}],

            [{item: 4,amount: 2,position:{x:3, y:6}},{item: 8,amount: 2,position:{x:3, y:7}},{item: 9,amount: 1,position:{x:3, y:8}},{item:"合成",position:{x:3,y:9}}],

            [{item: 4,amount: 2,position:{x:3, y:7}},{item: 8,amount: 2,position:{x:3, y:8}},{item: 9,amount: 1,position:{x:3, y:9}},{item:"合成",position:{x:3,y:10}}]

        ],

        [

            [{item: 10,amount: 1,position:{x:3, y:3}},{item: 13,amount: 1,position:{x:3, y:4}},{item:"合成",position:{x:3,y:5}}],

            [{item: 38,amount: 10,position:{x:3, y:4}},{item: 4,amount: 2,position:{x:3, y:5}},{item:"合成",position:{x:3,y:6}}],

            [{item: 7,amount: 8,position:{x:3, y:5}},{item: 13,amount: 2,position:{x:3, y:6}},{item:"合成",position:{x:3,y:7}}],

            [{item: 4,amount: 2,position:{x:3, y:6}},{item: 8,amount: 1,position:{x:3, y:7}},{item: 13,amount: 1,position:{x:3, y:8}},{item:"合成",position:{x:3,y:9}}],

            [{item: 38,amount: 1,position:{x:3, y:7}},{item: 4,amount: 1,position:{x:3, y:8}},{item: 11,amount: 1,position:{x:3, y:9}},{item:"合成",position:{x:3,y:10}}]

        ],

        [

            [{item: 25,amount: 1,position:{x:3, y:5}},{item: 44,amount: 1,position:{x:3, y:6}},{item:"合成",position:{x:3,y:7}}],

            [{item: 5,amount: 1,position:{x:3, y:6}},{item: 29,amount: 1,position:{x:3, y:7}},{item: 30,amount: 1,position:{x:3, y:8}},{item: 14,amount: 1,position:{x:3, y:9}},{item:"合成",position:{x:3,y:10}}],

            [{item: 25,amount: 1,position:{x:3, y:7}},{item: 32,amount: 1,position:{x:3, y:8}},{item:"合成",position:{x:3,y:9}}],

            [{item: 25,amount: 1,position:{x:3, y:8}},{item: 35,amount: 2,position:{x:3, y:9}},{item:"合成",position:{x:3,y:10}}],

        ],

        [

            [{item: 4,amount: 2,position: {x:3,y:6}},{item: 9,amount: 2,position: {x:3,y:7}},{item: 8,amount: 1,position: {x:3,y:8}},{item: 13,amount: 2,position: {x:3,y:9}},{item:"合成",position:{x:3,y:10}}],

            [{item: 4,amount: 2,position: {x:3,y:7}},{item: 38,amount: 2,position: {x:3,y:8}},{item:"合成",position:{x:3,y:9}}],

            [{item: 2,amount: 6,position: {x:3,y:8}},{item: 13,amount: 3,position: {x:3,y:9}},{item: 4,amount: 4,position: {x:3,y:10}},{item:"合成",position:{x:3,y:11}}],

            [{item: 7,amount: 2,position: {x:3,y:9}},{item: 38,amount: 2,position: {x:3,y:10}},{item:"合成",position:{x:3,y:11}}]

        ],

        [

            [{item: 38,amount: 3,position: {x:3,y:8}},{item:"合成",position:{x:3,y:9}}],

            [{item: 12,amount: 3,position: {x:3,y:9}},{item:"合成",position:{x:3,y:10}}]

        ]

    ];

    this.positionChange = function(position){

        return {x:position.x\*64,y:position.y\*64};

    }

    this.draw = function(ctx){

        ctx.beginPath();

        ctx.rect(5, 23, 316-12, 577-23);

        ctx.fillStyle = "#BEBEBE";

        ctx.fill();

        ctx.textAlign = 'center';

        // 第一層工具列

        for(var i = 0;i < this.synthesisBar.length; i++){

            this.backpack.position = this.positionChange(this.synthesisBar[i].position);

            this.backpack.draw(ctx);

            ctx.font = "15px Arial";

            ctx.fillStyle = "black";

            ctx.fillText(this.synthesisBar[i].item, this.synthesisBar[i].position.x\*64, this.synthesisBar[i].position.y\*64);

        }

        //第二層工具列

        if(this.firstColumeIndex != -1){

            for(var i = 0;i < this.synthesisBarDetail[this.firstColumeIndex].length; i++){

                this.backpack.position = this.positionChange(this.synthesisBarDetail[this.firstColumeIndex][i].position);

                this.backpack.draw(ctx);

                ctx.font = "15px Arial";

                ctx.fillStyle = "black";

                var index = this.synthesisBarDetail[this.firstColumeIndex][i].item-1;

                this.drawPicture(this.synthesisBarDetail[this.firstColumeIndex][i],ctx);

            }

        }

        //第三層工具列

        if(this.firstColumnIndex != -1 && this.secondColumnIndex != -1){

            for(var i = 0;i < this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex].length; i++){

                this.backpack.position = this.positionChange(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].position);

                this.backpack.draw(ctx);

                ctx.font = "15px Arial";

                ctx.fillStyle = "black";

                var check = backpackList.checkIfSynthesisAvailable(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex], this.synthesisBarDetail[this.firstColumeIndex][this.secondColumnIndex].item);

                if(i == this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex].length-1){

                    if(check == 1)

                        ctx.fillText("合成",this.backpack.position.x, this.backpack.position.y+5);

                    else if(check == -1)

                        ctx.fillText("沒空間拉",this.backpack.position.x, this.backpack.position.y+5);

                    else

                        ctx.fillText("沒材料拉",this.backpack.position.x, this.backpack.position.y+5);

                }else{

                    this.drawPicture(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i],ctx);

                    ctx.font = "30px Arial";

                    ctx.fillStyle = "white";

                    ctx.fillText(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].amount.toString(), this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].position.x\*64, this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].position.y\*64+15);

                    ctx.strokeStyle = 'blue';

                    ctx.lineWidth = 2.5;

                    ctx.strokeText(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].amount.toString(), this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].position.x\*64, this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].position.y\*64+15);

                }

            }

        }

    }

    this.drawPicture = function(item\_to\_draw,ctx){

        switch(item\_to\_draw.item){

            case 1:

                this.item\_flower.position = this.positionChange(item\_to\_draw.position);

                this.item\_flower.draw(ctx);

                break;

            case 2:

                this.item\_spider\_web.position = this.positionChange(item\_to\_draw.position);

                this.item\_spider\_web.draw(ctx);

                break;

            case 3:

                this.item\_stone.position = this.positionChange(item\_to\_draw.position);

                this.item\_stone.draw(ctx);

                break;

            case 4:

                this.item\_branch.position = this.positionChange(item\_to\_draw.position);

                this.item\_branch.draw(ctx);

                break;

            case 5:

                this.item\_waikei\_homework.position = this.positionChange(item\_to\_draw.position);

                this.item\_waikei\_homework.draw(ctx);

                break;

            case 6:

                this.item\_grass.position = this.positionChange(item\_to\_draw.position);

                this.item\_grass.draw(ctx);

                break;

            case 7:

                this.item\_wood.position = this.positionChange(item\_to\_draw.position);

                this.item\_wood.draw(ctx);

                break;

            case 8:

                this.item\_flint.position = this.positionChange(item\_to\_draw.position);

                this.item\_flint.draw(ctx);

                break;

            case 9:

                this.item\_gold.position = this.positionChange(item\_to\_draw.position);

                this.item\_gold.draw(ctx);

                break;

            case 10:

                this.item\_pigskin.position = this.positionChange(item\_to\_draw.position);

                this.item\_pigskin.draw(ctx);

                break;

            case 11:

                this.item\_bee\_sting.position = this.positionChange(item\_to\_draw.position);

                this.item\_bee\_sting.draw(ctx);

                break;

            case 12:

                this.item\_snow\_ball.position = this.positionChange(item\_to\_draw.position);

                this.item\_snow\_ball.draw(ctx);

                break;

            case 13:

                this.item\_rope.position = this.positionChange(item\_to\_draw.position);

                this.item\_rope.draw(ctx);

                break;

            case 14:

                this.item\_droplet.position = this.positionChange(item\_to\_draw.position);

                this.item\_droplet.draw(ctx);

                break;

            case 15:

                this.item\_pixilart.position = this.positionChange(item\_to\_draw.position);

                this.item\_pixilart.draw(ctx);

                break;

            case 16:

                this.item\_ax.position = this.positionChange(item\_to\_draw.position);

                this.item\_ax.draw(ctx);

                break;

            case 17:

                this.item\_fishing\_rod.position = this.positionChange(item\_to\_draw.position);

                this.item\_fishing\_rod.draw(ctx);

                break;

            case 18:

                this.item\_shovel.position = this.positionChange(item\_to\_draw.position);

                this.item\_shovel.draw(ctx);

                break;

            case 19:

                this.item\_gold\_ax.position = this.positionChange(item\_to\_draw.position);

                this.item\_gold\_ax.draw(ctx);

                break;

            case 20:

                this.item\_gold\_shovel.position = this.positionChange(item\_to\_draw.position);

                this.item\_gold\_shovel.draw(ctx);

                break;

            case 21:

                this.item\_gold\_pixilart.position = this.positionChange(item\_to\_draw.position);

                this.item\_gold\_pixilart.draw(ctx);

                break;

            case 22:

                this.item\_helmat.position = this.positionChange(item\_to\_draw.position);

                this.item\_helmat.draw(ctx);

                break;

            case 23:

                this.item\_armor.position = this.positionChange(item\_to\_draw.position);

                this.item\_armor.draw(ctx);

                break;

            case 24:

                this.item\_wood\_armor.position = this.positionChange(item\_to\_draw.position);

                this.item\_wood\_armor.draw(ctx);

                break;

            case 25:

                this.item\_spear.position = this.positionChange(item\_to\_draw.position);

                this.item\_spear.draw(ctx);

                break;

            case 26:

                this.item\_arror.position = this.positionChange(item\_to\_draw.position);

                this.item\_arror.draw(ctx);

                break;

            case 27:

                this.item\_king\_wand.position = this.positionChange(item\_to\_draw.position);

                this.item\_king\_wand.draw(ctx);

                break;

            case 28:

                this.item\_space\_wand.position = this.positionChange(item\_to\_draw.position);

                this.item\_space\_wand.draw(ctx);

                break;

            case 29:

                this.item\_fire\_wand.position = this.positionChange(item\_to\_draw.position);

                this.item\_fire\_wand.draw(ctx);

                break;

            case 30:

                this.item\_ice\_wand.position = this.positionChange(item\_to\_draw.position);

                this.item\_ice\_wand.draw(ctx);

                break;

            case 31:

                this.item\_lamp.position = this.positionChange(item\_to\_draw.position);

                this.item\_lamp.draw(ctx);

                break;

            case 32:

                this.item\_firebundle.position = this.positionChange(item\_to\_draw.position);

                this.item\_firebundle.draw(ctx);

                break;

            case 33:

                this.item\_camp.position = this.positionChange(item\_to\_draw.position);

                this.item\_camp.draw(ctx);

                break;

            case 34:

                this.item\_campfire.position = this.positionChange(item\_to\_draw.position);

                this.item\_campfire.draw(ctx);

                break;

            case 35:

                this.item\_ice.position = this.positionChange(item\_to\_draw.position);

                this.item\_ice.draw(ctx);

                break;

            case 36:

                this.item\_bush.position = this.positionChange(item\_to\_draw.position);

                this.item\_bush.draw(ctx);

                break;

            case 37:

                this.item\_flower\_picked.position = this.positionChange(item\_to\_draw.position);

                this.item\_flower\_picked.draw(ctx);

                break;

            case 38:

                this.item\_grass\_picked.position = this.positionChange(item\_to\_draw.position);

                this.item\_grass\_picked.draw(ctx);

                break;

            case 44:

                this.item\_bush\_growed\_dig.position = this.positionChange(item\_to\_draw.position);

                this.item\_bush\_growed\_dig.draw(ctx);

                break;

        }

    }

    this.updateChildBar = function(index){

        this.currentPoint = index;

        if(this.currentPoint.x == 1 &&  this.currentPoint.y <= 8 &&  this.currentPoint.y >= 4){

            this.firstColumeIndex = this.currentPoint.y - 4;

            this.secondColumnRange = [];

            this.secondColumnIndex = -1;

            this.thirdColumnRange = [];

            for(var i = 0;i < this.synthesisBarDetail[this.firstColumeIndex].length;i++){

                this.secondColumnRange.push(this.synthesisBarDetail[this.firstColumeIndex][i].position);

            }

        }else if(this.currentPoint.x == 2 && this.checkSecondColumn() != -1){

            this.secondColumnIndex = this.checkSecondColumn();

            this.thirdColumnRange = [];

            for(var i = 0; i < this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex].length;i++){

                this.thirdColumnRange.push(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][i].position);

            }

        }else if(this.currentPoint.x == 3 && this.checkThirdColumn()!=-1){

            this.mouseHover();

        }else{

            this.firstColumeIndex = -1;

            this.secondColumnIndex = -1;

            this.secondColumnRange = [];

            this.thirdColumnRange = [];

        }

    }

    this.mouseHover = function(){

        console.log("mouseHover");

    }

    this.checkSecondColumn = function(){

        for(var i = 0;i < this.secondColumnRange.length;i++){

            if(this.currentPoint.x == this.secondColumnRange[i].x && this.currentPoint.y == this.secondColumnRange[i].y)

                return i;

        }

        return -1;

    }

    this.checkThirdColumn = function(){

        for(var i = 0;i < this.thirdColumnRange.length;i++){

            if(this.currentPoint.x == this.thirdColumnRange[i].x && this.currentPoint.y == this.thirdColumnRange[i].y)

                return i;

        }

        return -1;

    }

    this.click = function(e){

        if(this.firstColumeIndex != -1 && this.secondColumnIndex != -1){

            var check = backpackList.checkIfSynthesisAvailable(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex]);

            var position = this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex][this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex].length-1].position;

            if(this.currentPoint.x == position.x && this.currentPoint.y == position.y){

                if(check == 1){

                    backpackList.addItemBySynthesis(this.synthesisBarDetail[this.firstColumeIndex][this.secondColumnIndex].item);

                    backpackList.update(this.synthesisBarMaterial[this.firstColumeIndex][this.secondColumnIndex]);

                    score.scoreAddBySynthesis();

                }

            }

        }

    }

    this.mousemove = function(e){

        this.mousePosition = e;

        var m\_position = {x:-1,y:-1};

        if(e.x >= 33 && e.x < 91)

            m\_position.x = 1;

        if(e.x >= 91 && e.x < 158)

            m\_position.x = 2;

        if(e.x >= 158 && e.x < 220)

            m\_position.x = 3;

        if(e.y >= 33 && e.y < 93)

            m\_position.y = 1;

        if(e.y >= 93 && e.y < 156)

            m\_position.y = 2;

        if(e.y >= 156 && e.y < 222)

            m\_position.y = 3;

        if(e.y >= 222 && e.y < 287)

            m\_position.y = 4;

        if(e.y >= 287 && e.y < 347)

            m\_position.y = 5;

        if(e.y >= 347 && e.y < 410)

            m\_position.y = 6;

        if(e.y >= 410 && e.y < 476)

            m\_position.y = 7;

        if(e.y >= 476 && e.y < 540)

            m\_position.y = 8;

        if(e.y >= 540 && e.y < 606)

            m\_position.y = 9;

        if(e.y >= 606 && e.y < 665)

            m\_position.y = 10;

        if(e.y >= 665 && e.y < 727)

            m\_position.y = 11;

        this.updateChildBar(m\_position);

    }

};

1. tool.js

var Tool = function(option) {

    this.constructor = function(constructOption){

        this.attack\_point = constructOption.attackPoint;

        this.tool = constructOption.imagePath;

        this.item\_num = constructOption.itemNumber;

        this.reduceDurability = constructOption.reduceDurability;

    }

    this.constructor(option);

    this.tool.scale = 0.8;

    this.mapPosition = {x:0, y:0};

    this.type = "equipment";

    this.place = "hand";

    this.durability = 100;

    this.magic\_attack\_point = 0;

    this.arror\_attack\_point = 0;

    this.status = true;

    this.isRegenerate = false;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.tool.draw(ctx);

    }

};

Object.defineProperty(Tool.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.tool.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. weapon.js

var Weapon = function(option) {

    this.constructor = function(constructOption){

        this.attack\_point = constructOption.attack[0];

        this.magic\_attack\_point = constructOption.attack[1];

        this.arror\_attack\_point = constructOption.attack[2];

        this.weapon = constructOption.imagePath;

        this.item\_num = constructOption.itemNumber;

        this.reduceDurability = constructOption.reduceDurability;

        this.weapon.scale = constructOption.scale;

    }

    this.constructor(option);

    this.mapPosition = {x:0, y:0};

    this.type = "equipment";

    this.place = "hand";

    this.durability = 100;

    this.amount = 1;

    this.status = true;

    this.isRegenerate = false;

    this.item\_can\_be\_picked = true;

    this.draw = function(ctx){

        this.weapon.draw(ctx);

    }

};

Object.defineProperty(Weapon.prototype, 'position', {

    get: function() {

        return this.mapPosition;

    },

    set: function(newValue) {

        this.mapPosition = newValue;

        this.weapon.position = {x: this.mapPosition.x \* 64, y: this.mapPosition.y \* 64};

    }

});

1. world\_map.js

var World\_map = function()

{

    this.demo\_dead\_trigger = true;

    this.item\_map\_Array = [];

    this.load = function(){

        this.item\_black\_berry = new Framework.Sprite(define.materialPath + 'item\_black\_berry.png');

        this.monster\_boss\_tocan = new Framework.Sprite(define.materialPath + 'boss\_tocan.png');

        this.monster\_tocan = new Framework.Sprite(define.materialPath + 'pig\_tocan.png');

        this.monster\_tocan = new Framework.Sprite(define.materialPath + 'cow\_tocan.png');

        this.monster\_tocan = new Framework.Sprite(define.materialPath + 'bat\_tocan.png');

        this.monster\_tocan = new Framework.Sprite(define.materialPath + 'bee\_tocan.png');

        this.item\_sapling = new Framework.Sprite(define.materialPath + 'item\_sapling.png');

        this.terrain\_plain = [];

        this.terrain\_plain.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_plain.png'));

        this.terrain\_plain[0].scale = 2;

        this.terrain\_plain.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_plain.png'));

        this.terrain\_plain[1].scale = 2;

        this.terrain\_plain.push(new Framework.Sprite(define.imageNightPath + 'terrain\_plain.png'));

        this.terrain\_plain[2].scale = 2;

        this.terrain\_water = [];

        this.terrain\_water.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_water.png'));

        this.terrain\_water[0].scale = 2;

        this.terrain\_water.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_water.png'));

        this.terrain\_water[1].scale = 2;

        this.terrain\_water.push(new Framework.Sprite(define.imageNightPath + 'terrain\_water.png'));

        this.terrain\_water[2].scale = 2;

        this.terrain\_mountain = [];

        this.terrain\_mountain.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_mountain.png'));

        this.terrain\_mountain[0].scale = 2;

        this.terrain\_mountain.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_mountain.png'));

        this.terrain\_mountain[1].scale = 2;

        this.terrain\_mountain.push(new Framework.Sprite(define.imageNightPath + 'terrain\_mountain.png'));

        this.terrain\_mountain[2].scale = 2;

        this.terrain\_blood\_water = [];

        this.terrain\_blood\_water.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_blood\_water.png'));

        this.terrain\_blood\_water[0].scale = 2;

        this.terrain\_blood\_water.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_blood\_water.png'));

        this.terrain\_blood\_water[1].scale = 2;

        this.terrain\_blood\_water.push(new Framework.Sprite(define.imageNightPath + 'terrain\_blood\_water.png'));

        this.terrain\_blood\_water[2].scale = 2;

        this.terrain\_forest = [];

        this.terrain\_forest.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_forest.png'));

        this.terrain\_forest[0].scale = 2;

        this.terrain\_forest.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_forest.png'));

        this.terrain\_forest[1].scale = 2;

        this.terrain\_forest.push(new Framework.Sprite(define.imageNightPath + 'terrain\_forest.png'));

        this.terrain\_forest[2].scale = 2;

        this.terrain\_lava = [];

        this.terrain\_lava.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_lava.png'));

        this.terrain\_lava[0].scale = 2;

        this.terrain\_lava.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_lava.png'));

        this.terrain\_lava[1].scale = 2;

        this.terrain\_lava.push(new Framework.Sprite(define.imageNightPath + 'terrain\_lava.png'));

        this.terrain\_lava[2].scale = 2;

        this.terrain\_snow\_ground = [];

        this.terrain\_snow\_ground.push(new Framework.Sprite(define.imageMorningPath + 'terrain\_snow\_ground.png'));

        this.terrain\_snow\_ground[0].scale = 2;

        this.terrain\_snow\_ground.push(new Framework.Sprite(define.imageAfternoonPath + 'terrain\_snow\_ground.png'));

        this.terrain\_snow\_ground[1].scale = 2;

        this.terrain\_snow\_ground.push(new Framework.Sprite(define.imageNightPath + 'terrain\_snow\_ground.png'));

        this.terrain\_snow\_ground[2].scale = 2;

        this.item\_honey = new Framework.Sprite(define.materialPath + 'Honey.png');

        this.item\_meat = new Framework.Sprite(define.materialPath + 'Meat.png');

        this.item\_monster\_meat = new Framework.Sprite(define.materialPath + 'Monster\_Meat.png');

        this.item\_bat\_wing = new Framework.Sprite(define.materialPath + 'Batilisk\_Wing.png');

        this.monster\_bat = new Framework.AnimationSprite({url:define.materialPath + 'bat.png', col:3 , row:4 , loop:true , speed:12});

        this.item\_eyeball = new Framework.Sprite(define.materialPath + 'Eyeball.png');

        this.monster\_cow = new Framework.AnimationSprite({url:define.materialPath + 'cow.png', col:3 , row:4 , loop:true , speed:12});

        this.monster\_cute\_little\_eye = new Framework.AnimationSprite({url:define.materialPath + 'monster\_cute\_litter\_eye.png', col:3 , row:4 , loop:true , speed:12});

        this.monster\_pig = new Framework.AnimationSprite({url:define.materialPath + 'pig.png', col:3 , row:4 , loop:true , speed:12});

        this.monster\_bee = new Framework.AnimationSprite({url:define.materialPath + 'bee.png', col:3 , row:4 , loop:true , speed:12});

        this.monster\_boss = new Framework.AnimationSprite({url:define.materialPath + 'boss.png', col:3 , row:4 , loop:true , speed:12});

        this.creation\_blood\_status = new Creation\_blood\_status();

        this.music\_stop = false;

        this.clock = new Clock();

        this.game\_object\_detail = new Game\_object\_detail();

        this.item\_flower\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_flower\_growed\_dig.png');

        this.item\_blank = new Framework.Sprite(define.materialPath + 'item\_blank.png');

        this.Floral = new Framework.Sprite(define.materialPath + 'Floral.png');

        this.item\_grass\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_grass\_growed\_dig.png');

        this.item\_berry = new Framework.Sprite(define.materialPath + 'item\_berry.png');

        this.item\_flower\_picked = new Framework.Sprite(define.materialPath + 'item\_flower\_picked.png');

        this.item\_flower\_dig = new Framework.Sprite(define.materialPath + 'item\_flower\_dig.png');

        this.item\_bush\_dig = new Framework.Sprite(define.materialPath + 'item\_bush\_dig.png');

        this.monster\_cute\_litter\_eye\_tocan = new Framework.Sprite(define.materialPath + 'monster\_cute\_litter\_eye\_tocan.png');

        this.item\_grass\_dig = new Framework.Sprite(define.materialPath + 'item\_grass\_dig.png');

        this.item\_tree\_dig = new Framework.Sprite(define.materialPath + 'item\_tree\_dig.png');

        this.map\_item\_tree = new Framework.Sprite(define.materialPath + 'map\_item\_tree.png');

        this.map\_item\_tree\_growed = new Framework.Sprite(define.materialPath + 'map\_item\_tree\_growed.png');

        this.map\_item\_tree\_cutted = new Framework.Sprite(define.materialPath + 'map\_item\_tree\_cutted.png');

        this.url = define.skillAnimationPath + 'Recovery4.png';

        this.flying\_arror = new Framework.AnimationSprite({url:this.url, col:5 , row:4 , loop:true , speed:16});

        this.flying\_arror\_left = new Framework.Sprite(define.materialPath + 'arror\_left.png');

        this.flying\_arror\_right = new Framework.Sprite(define.materialPath + 'arror\_right.png');

        this.flying\_arror\_up = new Framework.Sprite(define.materialPath + 'arror\_up.png');

        this.flying\_arror\_down = new Framework.Sprite(define.materialPath + 'arror\_down.png');

        this.item\_fish = new Framework.Sprite(define.materialPath + 'item\_fish.png');

        this.item\_sapling\_growed\_dig = new Framework.Sprite(define.materialPath + 'item\_sapling\_growed\_dig.png');

        this.item\_sapling\_dig = new Framework.Sprite(define.materialPath + 'item\_sapling\_dig.png');

        this.player1 = new BombMan(define.materialPath + 'Actor.png', {down: {from: 0, to: 2}, left: {from:3, to: 5}, right: {from: 6, to: 8}, up: {from: 9, to: 11}});

        this.player1.canvasPosition = {x:13, y:7};

        this.player1.position = {x:10, y:1};

        this.floor = new Framework.Sprite(define.builldingPath + '石磚.png');

        this.floor.scale = 2;

        this.wall = new Framework.Sprite(define.builldingPath + '石牆.png');

        this.wall.scale = 2;

        this.npc1 = new Npc1(this);

        this.npc1.position = {x:49,y:47};

        this.npc2 = new Npc2(this);

        this.npc2.position = {x:20,y:15};

        this.npc\_event = new Npc\_event(this);

        this.monster = [];

        var boss =  new Monster\_boss(this);

        boss.position = {x:393,y:393};

        this.monster.push(boss);

        this.stopMonster = false;

        this.stopMonsterCounter = 0;

        this.score = new Score();

        this.synthesisBar = new SynthesisBar(this.player1.getBackPack(), this.score);

        this.map\_selector = new Map\_selector();

        this.skill\_handler = new Skill\_handler();

        this.spear\_handler = new Spear\_handler();

        this.visitor = new ReduceDurabilityVisitor();

        this.creation\_blood\_status = new Creation\_blood\_status();

        this.fishing = new Fishing();

        this.objectFactory = new Object\_factory();

        this.handle\_initial\_character = new Handle\_initial\_character();

        this.playerInitial = false;

        this.audio = new Framework.Audio({

            kick: {

                mp3: define.musicPath + 'levelup.mp3',

            }, die\_scream:{

                mp3: define.musicPath + '女慘叫.mp3',

            }, monster\_attack:{

                mp3: define.musicPath + '怪物攻擊.mp3',

            }

        });

    }

    this.init = function()

    {

        this.game\_object\_detail.init();

        this.playerWalkDirection = {x:0, y:1};

        this.skillTimer = new Skill\_timer();

        this.character\_description = new Character\_description();

        this.arror\_attack = new Arror\_attack(this.player1, this.monster);

        this.handle\_initial\_character.init();

        this.constants = new Constants();

        this.is\_character\_description\_open = false;

        this.tileArray = [];

        this.tileArrayPosition = [];

        this.tilePosition = [];

        this.itemArray = [];

        this.mapArray = [];

        this.on\_map\_name = "World";

        this.clear = false;

        //playerPositionOnMap為人物出現在mapArray的位置，只要改這個，勿動其他常數

        this.playerPositionOnMap = {x:47,y:47};

        this.playerPositionOnMapSave = {

            "World": {x:47,y:47},

            "House1": {x:19, y:34}

        };

        this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

        this.addMonsterRandom();

        this.map\_selector.nullClean();

        for(var i = 0; i < 11;i++){

            this.tileArrayPosition = [];

            for(var j = 0; j < 11; j++)

                this.tileArrayPosition.push({x: j+8, y: i+2});

            this.tilePosition.push(this.tileArrayPosition);

        }

        this.creation\_blood\_status.init(this.player1);

    };

    this.goToMap = function(map\_name){

        this.on\_map\_name = map\_name;

        this.playerPositionOnMap = this.playerPositionOnMapSave[map\_name];

        this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

    }

    this.notifyDraw = function(){

        m\_map.draw(Framework.Game.\_context);

    }

    this.setPlayerPosition = function(playerPosition){

        this.player1.position = playerPosition;

    }

    //怪物攻擊的速度

    this.monster\_kill\_timer = 0;

    this.update = function()

    {

        if(this.monster[0].isdead){

            this.gameClear();

            Framework.Game.goToLevel('gameOver');

        }

        if(this.on\_map\_name == "World"){

            if(this.playerInitial){

                if(this.player1.player\_state == "alive" || false){

                    this.checkIsDie();

                }

                this.skill\_handler.update();

                this.spear\_handler.update();

                this.arror\_attack.update();

                if(this.pressWalk === true && !this.player1.beCaught)

                {

                    if(this.player1.player\_state == "alive" && this.checkIsWalkAble(this.playerWalkDirection))

                    {

                        if(this.keyPress == "Down") {

                            this.player1.walk({x:0,y:1});

                            this.playerPositionOnMap.y+=1;

                            this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.addMonsterRandom();

                            this.map\_selector.nullClean();

                        }

                        if(this.keyPress == "Left") {

                            this.player1.walk({x:-1,y:0});

                            this.playerPositionOnMap.x-=1;

                            this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.addMonsterRandom();

                            this.map\_selector.nullClean();

                        }

                        if(this.keyPress == "Right") {

                            this.player1.walk({x:1,y:0});

                            this.playerPositionOnMap.x+=1;

                            this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.addMonsterRandom();

                            this.map\_selector.nullClean();

                        }

                        if(this.keyPress == "Up") {

                            this.player1.walk({x:0,y:-1});

                            this.playerPositionOnMap.y-=1;

                            this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                            this.addMonsterRandom();

                            this.map\_selector.nullClean();

                        }

                    }

                }

                if(this.skillTimer.isEnergyFull){

                    this.skill\_handler.start(this.playerWalkDirection, this.playerPositionOnMap, this.player1.handEquipmentId);

                }

                this.player1.update();

                this.character\_description.update(this.player1);

                var hurt\_point=0;

                for(var i=0;i<this.monster.length;i++){

                    this.monster[i].update();

                    if(this.monster[i].isAttack()){

                        hurt\_point += this.monster[i].attack;

                    }

                }

                var i = 0;

                while(i < this.monster.length) {

                    if(this.monster[i].isdead){

                        if(this.map\_selector.checkIsBlank(this.on\_map\_name, this.monster[i].position)){

                            this.map\_selector.addObject(this.on\_map\_name, this.monster[i].position, this.monster[i].drop());

                            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                            m\_map.draw(Framework.Game.\_context);

                        }

                        this.monster.splice(i, 1);

                        this.player1.getExperience(5);

                        this.score.scoreAddByKillMonster();

                    }

                    else if(this.monster[i].health <= 0){

                        this.monster[i].die();

                        i++;

                    }else{

                        i++;

                    }

                }

                if(hurt\_point != 0)

                    this.player1GotHurt(hurt\_point);

                if(this.fishing.is\_start){

                    this.fishing.update();

                    if(this.player1.mode != "fishing"){

                        this.fishing.stop();

                        m\_map.draw(Framework.Game.\_context);

                    }

                }

                this.creation\_blood\_status.characterBloodUpdate(this.player1);

                this.creation\_blood\_status.characterMagicUpdate(this.player1);

                this.creation\_blood\_status.characterHungryUpdate(this.player1);

                this.creation\_blood\_status.monsterUpdate(this.monster);

                this.npc1.update();

                this.npc\_event.update();

            }else{

                this.handle\_initial\_character.update();

                if(this.handle\_initial\_character.is\_initial){

                    this.playerInitial = true;

                    this.player1.init();

                    this.player1.setCapibility(this.handle\_initial\_character.character\_description.character\_descruption\_point);

                    this.clockDraw(Framework.Game.\_context);

                    m\_map.draw(Framework.Game.\_context);

                }

            }

        }else if(this.on\_map\_name == "House1"){

            if(this.player1.player\_state == "alive")

                this.checkIsDie();

            this.skill\_handler.update();

            this.spear\_handler.update();

            this.arror\_attack.update();

            if(this.pressWalk === true && !this.player1.beCaught)

            {

                if(this.player1.player\_state == "alive" && this.checkIsWalkAble(this.playerWalkDirection))

                {

                    if(this.keyPress == "Down") {

                        this.player1.walk({x:0,y:1});

                        this.playerPositionOnMap.y+=1;

                        this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.addMonsterRandom();

                        this.map\_selector.nullClean();

                    }

                    if(this.keyPress == "Left") {

                        this.player1.walk({x:-1,y:0});

                        this.playerPositionOnMap.x-=1;

                        this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.addMonsterRandom();

                        this.map\_selector.nullClean();

                    }

                    if(this.keyPress == "Right") {

                        this.player1.walk({x:1,y:0});

                        this.playerPositionOnMap.x+=1;

                        this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.addMonsterRandom();

                        this.map\_selector.nullClean();

                    }

                    if(this.keyPress == "Up") {

                        this.player1.walk({x:0,y:-1});

                        this.playerPositionOnMap.y-=1;

                        this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                        this.addMonsterRandom();

                        this.map\_selector.nullClean();

                    }

                }

            }

            if(this.skillTimer.isEnergyFull){

                this.skill\_handler.start(this.playerWalkDirection, this.playerPositionOnMap, this.player1.handEquipmentId);

            }

            this.player1.update();

            this.npc2.update();

            this.npc\_event.update();

            this.character\_description.update(this.player1);

            var hurt\_point=0;

            this.creation\_blood\_status.characterBloodUpdate(this.player1);

            this.creation\_blood\_status.characterMagicUpdate(this.player1);

            this.creation\_blood\_status.characterHungryUpdate(this.player1);

        }

    }

    this.player1GotHurt = function(hurt\_point) {

        this.monster\_kill\_timer ++;

        var audio = new Framework.Audio({

            monster\_attack:{

                mp3: define.musicPath + 'monster\_attack.mp3',

        }

        });

        if(this.monster\_kill\_timer == 15){

            this.player1.gethurt(hurt\_point);

            this.monster\_kill\_timer = 0;

            audio.play({name: 'monster\_attack', loop: false});

        }

    }

    this.draw = function(ctx) {

        if(this.playerInitial){

            for(var i=0; i<11; i++){

                for(var j=0; j<11; j++){

                    switch(this.mapArray[j][i]){

                        case 192:

                            this.terrain\_plain[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_plain[this.clock.status].draw(ctx);

                            break;

                        case 200:

                            this.terrain\_water[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_water[this.clock.status].draw(ctx);

                            break;

                        case 137:

                            this.terrain\_mountain[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_mountain[this.clock.status].draw(ctx);

                            break;

                        case 91:

                            this.terrain\_blood\_water[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_blood\_water[this.clock.status].draw(ctx);

                            break;

                        case 123:

                            this.terrain\_forest[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_forest[this.clock.status].draw(ctx);

                            break;

                        case 196:

                            this.terrain\_lava[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_lava[this.clock.status].draw(ctx);

                            break;

                        case 255:

                            this.terrain\_snow\_ground[this.clock.status].position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.terrain\_snow\_ground[this.clock.status].draw(ctx);

                            break;

                        case 3:

                            this.floor.position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.floor.draw(ctx);

                            break;

                        case 4:

                            this.wall.position = {x:this.tilePosition[j][i].x\*64,y:this.tilePosition[j][i].y\*64};

                            this.wall.draw(ctx);

                            break;

                    }

                }

            }

            this.player1.draw(ctx);

            for(var i=0; i<11; i++)

                for(var j=0; j<11; j++){

                    this.itemArray[j][i].position = {x:this.tilePosition[j][i].x,y:this.tilePosition[j][i].y};

                    this.itemArray[j][i].draw(ctx);

                }

            ctx.beginPath();

            ctx.rect(1185, 100, 80, 700);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            ctx.beginPath();

            ctx.rect(380, 80, 100, 700);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            ctx.beginPath();

            ctx.rect(280, 800, 1100, 90);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            if(this.skillTimer.buttonPress)

                this.skillTimer.draw(ctx);

            this.arror\_attack.draw(ctx);

            if(this.fishing.is\_start)

                this.fishing.draw(ctx);

            this.clock.draw(ctx);

            for(var i=0;i<this.monster.length;i++){

                this.monster[i].draw(ctx);

            }

            if(this.skill\_handler.isAnimationStart()){

                for(var i=-5,ii=0; i<6; i++,ii++){

                    for(var j=-5,jj=0; j<6; j++,jj++){

                        if(this.skill\_handler.mapPosition.x == i + this.playerPositionOnMap.x && this.skill\_handler.mapPosition.y == j+ this.playerPositionOnMap.y){

                            this.skill\_handler.setPosition({x:64\*this.tilePosition[jj][ii].x,y:64\*this.tilePosition[jj][ii].y});

                            this.skill\_handler.draw(ctx);

                        }

                    }

                }

            }else if(this.spear\_handler.spear.\_start){

                for(var i=-5,ii=0; i<6; i++,ii++){

                    for(var j=-5,jj=0; j<6; j++,jj++){

                        if(this.spear\_handler.mapPosition.x == i + this.playerPositionOnMap.x && this.spear\_handler.mapPosition.y == j+ this.playerPositionOnMap.y){

                            this.spear\_handler.spear.position = {x:64\*this.tilePosition[jj][ii].x,y:64\*this.tilePosition[jj][ii].y};

                            this.spear\_handler.draw(ctx);

                        }

                    }

                }

            }

            this.game\_object\_detail.draw(ctx);

            ctx.beginPath();

            ctx.rect(242, 256, 225, 600);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            ctx.beginPath();

            ctx.rect(480, 0, 700, 96);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            ctx.beginPath();

            ctx.rect(1180, 426, 414, 470);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            ctx.beginPath();

            ctx.rect(258, 870, 1257, 26);

            ctx.fillStyle = "#BEBEBE";

            ctx.fill();

            this.synthesisBar.draw(ctx);

            this.npc1.draw(ctx);

            this.npc2.draw(ctx);

            ctx.font = "20px Arial";

            ctx.fillStyle = "black";

            ctx.textAlign = 'center';

            this.player1.backpack.draw(ctx);

this.player1.equipmentBar.draw(ctx);

            if(this.handle\_initial\_character.name == "請輸入名字"){

                this.handle\_initial\_character.name = "";

            }

            if(this.npc\_event.taking\_is\_start){

                ctx.beginPath();

                ctx.rect(201, 98, 272, 196);

                ctx.fillStyle = "#BEBEBE";

                ctx.fill();

            }

            if(!this.is\_character\_description\_open){

                this.npc\_event.draw(ctx);

            }

            if(!this.npc\_event.taking\_is\_start)

                this.character\_description.draw(ctx);

            if(!this.npc\_event.taking\_is\_start){

                this.creation\_blood\_status.draw(ctx);

                ctx.fillText(this.handle\_initial\_character.name, 252 ,250);

            }

        }else

this.handle\_initial\_character.draw(ctx);

    }

    this.clockDraw = function(ctx){

        this.clock.init();

        var clockInterval = setInterval(() => {

            if(!this.npc\_event.taking\_is\_start){

                this.clock.draw(ctx);

                this.creation\_blood\_status.draw(ctx);

            }

            if(this.player1.character\_descruption\_total\_point[0] <= 0 || this.clear){

                clearInterval(clockInterval);

            }

        }, 500);

    }

    this.checkKeyIsPress  = function(key){

        for(var i=0;i<this.capture\_key.length;i++){

            if(this.capture\_key[i].key == key){

                return true;

            }

        }

        return false;

    }

    this.whatIsTheLastKeyMove  = function(){

        for(var i=this.capture\_key.length-1;i>=0;i--){

            if(this.capture\_key[i].key == 'Down' || this.capture\_key[i].key == 'Up' ||

                this.capture\_key[i].key == 'Left' || this.capture\_key[i].key == 'Right'){

                return this.capture\_key[i].key;

                break;

            }

        }

        return "No";

    }

    this.addMonsterRandom = function(){

        var map\_number,flag\_x,flag\_y;

        if(this.map\_selector.local\_map\_0.proxy.null\_map.length > 0){

            for(var i=0;i<this.map\_selector.local\_map\_0.proxy.null\_map.length;i++){

                map\_number = this.map\_selector.local\_map\_0.proxy.null\_map[i];

                flag\_x = Math.floor(map\_number/10)\*40;

                flag\_y = (map\_number%10)\*40;

                this.createMonster(flag\_x, flag\_y);

            }

        }

    }

    this.createMonster = function(flag\_x, flag\_y){

        var amount = Math.floor(Math.random()\*5) + 5;

        var count = 0;

        while(count != amount){

            var position\_x = (Math.floor(Math.random()\*40));

            var position\_y = (Math.floor(Math.random()\*40));

            var m\_position = {x: flag\_x + position\_x,y: flag\_y + position\_y};

            if(this.map\_selector.checkFloorCanWalk(this.on\_map\_name, m\_position) && this.map\_selector.checkIsBlank(this.on\_map\_name, m\_position)){

                var newMonster = this.objectFactory.createMonster(this);

                newMonster.position = m\_position;

                this.monster.push(newMonster);

                count++;

            }

        }

    }

    var m\_map = this;

    this.deadClear = function(){

        this.skillTimer.clear();

        this.capture\_key = [];

        this.player1.character\_descruption\_point[0] = 0;

        this.clock.stopMusic(true);

        this.audio.stopAll();

        this.player1.update();

        this.clock.stopMusic(true);

    }

    this.gameClear = function(){

        this.skillTimer.clear();

        this.audio.stopAll();

        this.capture\_key = [];

        this.clear = true;

        this.clock.stopMusic(true);

        this.player1.gameClear = true;

        this.clock.stopMusic(true);

    }

    this.checkIsDie = function(){

        if(this.player1.character\_descruption\_point[0] <= 0 && this.demo\_dead\_trigger){

            var audio = new Framework.Audio({

                die\_scream: {

                    mp3: define.musicPath + '女慘叫.mp3'

                }

            });

            this.player1.dieEvent({x: 13, y: 7});

            this.audio.play({name: 'die\_scream', loop: false});

            this.update();

            m\_map.draw(Framework.Game.\_context);

            setTimeout(()=>{

                this.deadClear();

                m\_map.player1.die();

            },4000);

        }

    }

    this.getLeftMonsterNum = function()

    {

        var count =0;

        for(var i=0;i<this.monster.length;i++)

            if(this.monster[i].isDead === false)

                count++;

        return count;

    }

    this.pressWalk = false;

    this.keyPress = "";

    this.key\_is\_press = false;

    this.skill\_number = -99;

    this.capture\_key = [];

    this.keydown = function(e, list){

        this.capture\_key.push(e);

        if(e.key != 'Space' && this.fishing.is\_start)

            this.fishing.stop();

        //Q:死亡結束  W:勝利結束 R:蜜蜂 T:豬 Y:牛 U:眼球 I:蝙蝠 O:boss

        switch(e.key){

            case 'Q':

                this.gameClear();

                Framework.Game.goToLevel('gameOver');

                break;

            case 'W':

                this.deadClear();

                this.audio.stopAll();

                Framework.Game.goToLevel('gameOver');

                break;

            case 'O':

                var newMonster1 =  new Monster\_boss(this);

                newMonster1.position = this.playerPositionOnMap;

                this.monster.push(newMonster1);

                break;

            case 'R':

                var newMonster1 =  new Monster\_bee(this);

                newMonster1.position = this.playerPositionOnMap;

                this.monster.push(newMonster1);

                break;

            case 'T':

                var newMonster1 =  new Monster\_pig(this);

                newMonster1.position = this.playerPositionOnMap;

                this.monster.push(newMonster1);

                break;

            case 'Y':

                var newMonster1 =  new Monster\_cow(this);

                newMonster1.position = this.playerPositionOnMap;

                this.monster.push(newMonster1);

                break;

            case 'U':

                var newMonster1 =  new Monster\_cute\_little\_eye(this);

                newMonster1.position = this.playerPositionOnMap;

                this.monster.push(newMonster1);

                break;

            case 'I':

                var newMonster1 =  new Monster\_bat(this);

                newMonster1.position = this.playerPositionOnMap;

                this.monster.push(newMonster1);

                break;

            case 'S':

                this.keyPress = "S";

                if(this.player1.mode == "magic"){

                    this.skillTimer.startAccumulateEnergy();

                    this.drawSkillTimer(Framework.Game.\_context);

                }

                break;

            case 'D':

                this.handleDrop();

                this.handleHoverBackpack();

                break;

            case 'P':

                this.demo\_dead\_trigger = false;

                break;

            case 'F':

                if(this.player1.mode == "fishing" && !this.fishing.is\_start)

                    this.startFishing();

                break;

            case 'E':

                if(this.character\_description.is\_character\_description\_open){

                    this.character\_description.is\_character\_description\_open = false;

                }else{

                    this.character\_description.is\_character\_description\_open = true;

                }

                break;

            case 'Z':

                if(this.npc\_event.taking\_is\_start){

                    this.npc\_event.amount -= 2;

                    if( this.npc\_event.amount <= -1){

                            this.npc\_event.amount = -1;

                        }

                    this.npc\_event.talking();

                    this.draw(Framework.Game.\_context);

                }

                break;

            case 'Space':

                this.handleSpace();

                this.handleHoverBackpack();

                m\_map.draw(Framework.Game.\_context);

                break;

            default:

                break;

        }

        if(this.player1.player\_state == "alive" && !this.npc\_event.taking\_is\_start){

            if(this.whatIsTheLastKeyMove() == 'Down'){

                if(this.playerPositionOnMap.x == 19 && this.playerPositionOnMap.y ==34 && this.on\_map\_name == "House1"){

                    this.playerPositionOnMapSave["House1"] = this.playerPositionOnMap;

                    this.playerPositionOnMap = this.playerPositionOnMapSave["World"];

                    this.goToMap("World");

                }

                this.player1.walk({x:0,y:1});

                this.playerWalkDirection = {x:0,y:1};

                this.keyPress = "Down";

                if(this.checkIsWalkAble(this.playerWalkDirection)){

                    this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.pressWalk = true;

                    this.addMonsterRandom();

                    this.map\_selector.nullClean();

                }

            }else if(this.whatIsTheLastKeyMove() == 'Left'){

                this.playerWalkDirection = {x:-1,y:0};

                this.player1.walk({x:-1,y:0});

                this.keyPress = "Left";

                if(this.checkIsWalkAble(this.playerWalkDirection)){

                    this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name,this.playerPositionOnMap);

                    this.pressWalk = true;

                    this.addMonsterRandom();

                    this.map\_selector.nullClean();

                }

            }else if(this.whatIsTheLastKeyMove() == 'Right'){

                this.playerWalkDirection = {x:1,y:0};

                this.player1.walk({x:1,y:0});

                this.keyPress = "Right";

                if(this.checkIsWalkAble(this.playerWalkDirection)){

                    this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.pressWalk = true;

                    this.addMonsterRandom();

                    this.map\_selector.nullClean();

                }

            }else if(this.whatIsTheLastKeyMove() == 'Up'){

                if(this.playerPositionOnMap.x == 66 && this.playerPositionOnMap.y ==58 && this.on\_map\_name == "World"){

                    this.playerPositionOnMapSave["World"] = this.playerPositionOnMap;

                    this.playerPositionOnMap = this.playerPositionOnMapSave["House1"];

                    this.goToMap("House1");

                }

                this.playerWalkDirection = {x:0,y:-1};

                this.player1.walk({x:0,y:-1});

                this.keyPress = "Up";

                if(this.checkIsWalkAble(this.playerWalkDirection)){

                    this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                    this.pressWalk = true;

                    this.addMonsterRandom();

                    this.map\_selector.nullClean();

                }

            }

        }

    }

    this.keyup = function(e, list){

        var audio = new Framework.Audio({

            fight: {

                mp3: define.musicPath + '打擊.mp3',

            },

            spear: {

                mp3: define.musicPath + '長矛.mp3',

            },

            arror: {

                mp3: define.musicPath + '弓箭.mp3',

            },fire: {

                mp3: define.musicPath + '火魔法.mp3',

            },

            ice: {

                mp3: define.musicPath + '冰魔法.mp3',

            },move: {

                mp3: define.musicPath + '順移.mp3'

            }

        });

        if(e.key == 'S'){

            var attackMode = new Null\_attack();

            if(this.player1.mode == "hide" && !this.player1.hide){

                this.player1.hidePlayer();

                this.player1.hideAnimation.start();

            }

            else if(this.player1.mode == "magic"){

                if(this.skillTimer.isEnergyFull){

                    if(this.player1.equipmentBar.getEquipment(2).item\_num == 29)

                        audio.play({name: 'fire', loop: false});

                    if(this.player1.equipmentBar.getEquipment(2).item\_num == 30)

                        audio.play({name: 'ice', loop: false});

                    attackMode = new Magic\_attack(this.player1, this.monster, this.skill\_handler.mapPosition);

                }

                this.skillTimer.stopAccumulateEnergy();

            }

            else if(this.player1.mode == "arror"){

                this.arror\_attack.setPositionAndDirection(this.playerWalkDirection, this.playerPositionOnMap);

                attackMode = this.arror\_attack;

                audio.play({name: 'arror', loop: false});

            }else if(this.player1.mode == "spear"){

                attackMode = new Normal\_attack(this.player1, this.monster, this.playerWalkDirection, this.playerPositionOnMap);

                this.spear\_handler.start(this.playerWalkDirection, this.playerPositionOnMap);

                audio.play({name: 'spear', loop: false});

            }else if(this.player1.mode == "space"){

                this.player1.transportPlayer();

                this.playerPositionOnMap =  {x:392,y:392};

                this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                this.mapArray = this.map\_selector.makeMap(this.on\_map\_name, this.playerPositionOnMap);

                audio.play({name: 'move', loop: false});

            }else{

                attackMode = new Normal\_attack(this.player1, this.monster, this.playerWalkDirection, this.playerPositionOnMap);

                audio.play({name: 'fight', loop: false});

            }

            this.player1.attack(attackMode);

        }

        for(var i=0;i<this.capture\_key.length;i++){

            if( this.capture\_key[i].key == e.key){

                this.capture\_key.splice(i, 1);

                break;

            }

        }

        if( !(this.checkKeyIsPress('Down') || this.checkKeyIsPress('Up') ||

            this.checkKeyIsPress('Left') || this.checkKeyIsPress('Right')) ) {

            {

                this.player1.walkEnd();

                this.pressWalk = false;

                this.keyPress = "";

            };

        }

        m\_map.draw(Framework.Game.\_context);

    }

    this.handleFishing = function(){

        this.fishing.stop();

        var addSuccess = false;

        for(var i=-1;i<2;i++){

            for(var j=-1;j<2;j++){

                if(this.mapArray[5+i][5+j] != 91 &&

                    this.mapArray[5+i][5+j] != 200 &&

                    this.itemArray[5+i][5+j].item\_num == 0

                    ){

                        if(i == 0 && j == 0)

                            continue;

                        this.map\_selector.addObject(this.on\_map\_name, {x:this.playerPositionOnMap.x+j, y:this.playerPositionOnMap.y+i}, new Item\_fish());

                        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                        m\_map.draw(Framework.Game.\_context);

                        addSuccess = true;

                        this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

                        break;

                }

            }

            if(addSuccess)

                break;

        }

    }

    this.startFishing = function(){

        var audio = new Framework.Audio({

            fish: {

                mp3: define.musicPath + '釣魚.mp3',

            }

        });

        if(this.mapArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x] == 200){

            audio.play({name: 'fish', loop: false});

            this.fishing.start(this.playerWalkDirection);

        }

    }

    this.handleDrop = function(){

        if(this.mapArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x] != 91 &&

            this.mapArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x] != 200){

            if(this.player1.equipmentBar.selectedIndex != -1 && this.player1.equipmentBar.getSelectedEquipment() != null){

                if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 0){

                    this.map\_selector.addObject(this.on\_map\_name, {x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, this.player1.equipmentBar.getSelectedEquipment());

                    this.player1.equipmentBar.dropSelectedEquipment();

                }

            }else if(this.player1.backpack.selectedIndex != -1 && this.player1.backpack.getSelectedItem() != null){

                if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 0){

                    this.map\_selector.addObject(this.on\_map\_name, {x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, this.player1.backpack.getSelectedItem());

                    this.player1.backpack.dropSelectedItem();

                }else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == this.player1.backpack.getSelectedItem().item\_num){

                    if(this.player1.backpack.getSelectedItem().type == "material" ||

                    this.player1.backpack.getSelectedItem().type == "food" ||

                    this.player1.backpack.getSelectedItem().type == "plant"){

                        this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].amount += this.player1.backpack.getSelectedItem().amount;

                        this.player1.backpack.dropSelectedItem();

                    }

                }

            }

            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

            m\_map.draw(Framework.Game.\_context);

        }

    }

    this.handleSpace = function(){

        if(this.playerPositionOnMap.x + this.playerWalkDirection.x == this.npc1.position.x &&  this.on\_map\_name == "World" &&

            this.playerPositionOnMap.y + this.playerWalkDirection.y == this.npc1.position.y ){

                if(!this.npc\_event.taking\_is\_start){

                    this.npc\_event.trigger(this.npc1.name, "dialog");

                    this.npc\_event.talking();

                }

                else

                    this.npc\_event.talking();

                this.draw(Framework.Game.\_context);

        }

        if(this.playerPositionOnMap.x + this.playerWalkDirection.x == this.npc2.position.x &&  this.on\_map\_name == "House1" &&

            this.playerPositionOnMap.y + this.playerWalkDirection.y == this.npc2.position.y ){

                if(!this.npc\_event.taking\_is\_start){

                    this.npc\_event.trigger(this.npc2.name, "dialog");

                    this.npc\_event.talking();

                }

                else{

                    this.npc\_event.talking();

                }

                this.draw(Framework.Game.\_context);

        }

        if(this.player1.mode == "fishing" && this.fishing.fishBeCaught){

            this.handleFishing();

        }

        if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num !=0){

            if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 3 ||

                this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == -3){

                if(this.player1.mode == "rock\_dig")

                    this.handleRockDig();

            }

            else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].isRegenerate && this.player1.mode == "plant\_dig"){

                this.handlePlantDig();

            }

            else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].status)

            {

                if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == -1

                    && (this.player1.mode == "cut\_tree") )

                    this.handleCutTree();

                if(this.player1.backpack.checkIfPickAvailable(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num)){

                    this.handlePick();

                }

            }

        }

        m\_map.draw(Framework.Game.\_context);

    }

    this.handlePlantDig = function(){

        var audio = new Framework.Audio({

            dig: {

                mp3: define.musicPath + '挖掘.mp3',

            }

        });

        if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 1){

            if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].status){

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_flower\_growed\_dig());

            }else{

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_flower\_dig());

            }

            this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        }else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 36){

            if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].status){

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_bush\_growed\_dig());

            }else{

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_bush\_dig());

            }

            this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        }else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 6){

            if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].status){

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_grass\_growed\_dig());

            }else{

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_grass\_dig());

            }

            this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        }else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == -4){

            if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].status){

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_sapling\_growed\_dig());

            }else{

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_sapling\_dig());

            }

            this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        }

        else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == -1 &&

            this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].treeStatus == 2){

            this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_tree\_dig());

            this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        }

        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

        m\_map.draw(Framework.Game.\_context);

        audio.play({name: 'dig', loop: false});

    }

    this.handlePick = function(){

        var audio = new Framework.Audio({

            pick: {

                mp3: define.musicPath + '撿取.mp3',

            }

        });

        if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 1){

            this.player1.backpack.addItemByObject(new Item\_flower\_picked());

            this.pickRegenerateObject();

        }

        else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 6){

            this.player1.backpack.addItemByObject(new Item\_grass\_picked());

            this.pickRegenerateObject();

        }

        else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == -4){

            this.player1.backpack.addItemByObject(new Item\_branch());

            this.pickRegenerateObject();

        }

        else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 36){

            this.player1.backpack.addItemByObject( new Item\_berry());

            this.pickRegenerateObject();

        }

        else if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num >= 0){

            this.player1.backpack.addItemByObject(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x]);

            this.pickObject();

            if(this.checkIsWalkAble(this.playerWalkDirection) && this.keyPress != "")

                this.pressWalk = true;

        }

        audio.play({name: 'pick', loop: false});

    }

    this.handleCutTree = function(){

        var audio = new Framework.Audio({

            cut: {

                mp3: define.musicPath + '砍樹.mp3',

            }

        });

        var x = 5+this.playerWalkDirection.x;

        var y = 5+this.playerWalkDirection.y;

        var count = false;

        this.itemArray[y][x].update();

        this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        if(this.itemArray[y][x].dropWood){

            for(var i=-1;i<2;i++){

                for(var j=-1;j<2;j++){

                    if( this.mapArray[y+j][x+i] != 91 &&

                        this.mapArray[y+j][x+i] != 200 &&

                        this.itemArray[y+j][x+i].item\_num == 0

                        ){

                        if(((y+j) != 5) || ((x+i) != 5)){

                            count = true;

                            this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+i+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y+j}, new Item\_wood());

                            this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                            m\_map.draw(Framework.Game.\_context);

                            break;

                        }

                    }

                }

                if(count)

                    break;

            }

        }

        if(this.itemArray[y][x].treeStatus == 2){

            if(this.player1.getExperience(8)){

                this.audio.play({name: 'kick', loop: false});

            }

        }

        audio.play({name: 'cut', loop: false});

    }

    this.handleRockDig = function(){

        var audio = new Framework.Audio({

            dig: {

                mp3: define.musicPath + '挖掘.mp3',

            }

        });

        var y = 5+this.playerWalkDirection.y;

        var x = 5+this.playerWalkDirection.x;

        if(this.itemArray[y][x].item\_num == 3){

            this.itemArray[y][x].update();

            if(this.itemArray[y][x].count == 5){

                var flint = new Item\_flint();

                flint.amount = Math.floor(Math.random()\*3) + 1;

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, flint);

                var gold = new Item\_gold();

                gold.amount = Math.floor(Math.random()\*3) + 1;

                this.map\_selector.addObject(this.on\_map\_name,{x:1 + this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, gold);

            }

        }

        else{

            this.itemArray[y][x - 1].update();

            if(this.itemArray[y][x - 1].count == 5){

                var flint = new Item\_flint();

                flint.amount = Math.floor(Math.random()\*3) + 1;

                this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, flint);

                var gold = new Item\_gold();

                gold.amount = Math.floor(Math.random()\*3) + 1;

                this.map\_selector.addObject(this.on\_map\_name,{x:-1 + this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, gold);

            }

        }

        this.player1.equipmentBar.equipmentList[2].reduceDurability(this.visitor);

        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

        m\_map.draw(Framework.Game.\_context);

        audio.play({name: 'dig', loop: false});

    }

    this.handlePlant = function(){

        var audio = new Framework.Audio({

            dig: {

                mp3: define.musicPath + '挖掘.mp3',

            }

        });

        if(this.mapArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x] != 91 &&

            this.mapArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x] != 200){

            if(this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].item\_num == 0){

                switch(this.player1.backpack.itemList[this.player1.plantIndex].item\_num){

                    case 40:

                        var bush = new Item\_bush();

                        bush.update();

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, bush);

                        this.regeneration\_time = this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].regeneration\_time;

                        this.callDrawRegenerationTime();

                        break;

                    case 41:

                        var flower = new Item\_flower();

                        flower.update();

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, flower);

                        this.regeneration\_time = this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].regeneration\_time;

                        this.callDrawRegenerationTime();

                        break;

                    case 42:

                        var tree = new Map\_item\_tree(this);

                        tree.treeStatus = 2;

                        //長兩階段

                        tree.tryGrow();

                        tree.tryGrow();

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, tree);

                        break;

                    case 43:

                        var grass = new Item\_grass();

                        grass.update();

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, grass);

                        this.regeneration\_time = this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].regeneration\_time;

                        this.callDrawRegenerationTime();

                        break;

                    case 44:

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_bush());

                        break;

                    case 45:

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_flower());

                        break;

                    case 46:

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_grass());

                        break;

                    case 47:

                        var sapling = new Item\_sapling();

                        sapling.update();

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, sapling);

                        this.regeneration\_time = this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].regeneration\_time;

                        this.callDrawRegenerationTime();

                        break;

                    case 48:

                        this.map\_selector.addObject(this.on\_map\_name,{x:this.playerPositionOnMap.x+this.playerWalkDirection.x, y:this.playerPositionOnMap.y+this.playerWalkDirection.y}, new Item\_sapling());

                        break;

                    default:

                        break;

                }

                this.player1.backpack.itemList[this.player1.plantIndex].amount -= 1;

                this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

                m\_map.draw(Framework.Game.\_context);

                if(this.player1.backpack.itemList[this.player1.plantIndex].amount == 0)

                    this.player1.backpack.arrayRemoveByIndex(this.player1.plantIndex);

            }

        }

        audio.play({name: 'dig', loop: false});

    }

    this.pickRegenerateObject = function(){

        this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].update();

        m\_map.draw(Framework.Game.\_context);

        this.regeneration\_time = this.itemArray[5+this.playerWalkDirection.y][5+this.playerWalkDirection.x].regeneration\_time;

        this.callDrawRegenerationTime();

    }

    this.callDrawRegenerationTime = function(){

        setTimeout(()=>{m\_map.draw(Framework.Game.\_context)},this.regeneration\_time);

    }

    this.pickObject = function(){

        this.map\_selector.pickObject(this.on\_map\_name, {x:this.playerPositionOnMap.x + this.playerWalkDirection.x ,y:this.playerPositionOnMap.y + this.playerWalkDirection.y})

        this.itemArray = this.map\_selector.makeItemMap(this.on\_map\_name, this.playerPositionOnMap);

        m\_map.draw(Framework.Game.\_context);

    }

    this.drawSkillTimer = function(ctx){

        var interval = setInterval(()=>{

            if(this.skillTimer.buttonPress)

                this.skillTimer.draw(ctx);

            else

                clearInterval(interval);

        },100);

    }

    this.checkIsWalkAble = function(direction){  //檢查人物是否超過地圖大小

        var x = 5+direction.x;

        var y = 5+direction.y;

        var xx = this.playerPositionOnMap.x+direction.x;

        var yy = this.playerPositionOnMap.y+direction.y;

        if(this.mapArray[y][x] == 91 || this.mapArray[y][x] == 200 || this.itemArray[y][x].item\_num !=0 || this.mapArray[y][x] == 4 ||

            (xx == this.npc1.position.x && yy == this.npc1.position.y && this.on\_map\_name == "World") ||

            (xx == this.npc2.position.x && yy == this.npc2.position.y && this.on\_map\_name == "House1")){

            return false;

        }else{

            return true;

        }

    }

    this.checkMonsterIsWalkAble = function(map\_position){  //檢查人物是否超過地圖大小

        if(map\_position.x == this.playerPositionOnMap.x && map\_position.y == this.playerPositionOnMap.y)

            return false;

        else if(this.map\_selector.checkFloorCanWalk(this.on\_map\_name, map\_position) && this.map\_selector.checkIsBlank(this.on\_map\_name, map\_position)){

            return true;

        }else{

            return false;

        }

    }

    this.click = function(e){

        console.log(e);

        if(this.playerInitial){

            if(this.character\_description.is\_character\_description\_open){

                if(this.player1.capabilityt\_point !=0){

                    this.player1.characterAbilityClick(e);

                }

            }

            this.synthesisBar.click(e);

            this.player1.click(e);

            if(this.player1.plantIndex != -1){

                this.handlePlant();

            }

            this.handleHoverBackpack();

        }else{

            this.handle\_initial\_character.click(e);

        }

        m\_map.draw(Framework.Game.\_context);

    }

    this.mousemove = function(e){

        if(this.playerInitial){

            this.synthesisBar.mousemove(e);

            this.player1.mousemove(e);

            this.handleHoverBackpack();

        }else{

            this.handle\_initial\_character.mousemove(e);

        }

    }

    this.handleHoverBackpack = function(){

        if(this.player1.backpack.getSelectedItem() != null){

            this.game\_object\_detail.showUpdate(this.player1.backpack.getSelectedItem().item\_num);

            m\_map.draw(Framework.Game.\_context);

        }

        else{

            this.game\_object\_detail.showUpdate(null);

            m\_map.draw(Framework.Game.\_context);

        }

    }

}