

## Final Report

Team name : oops!!

Team member : B06902026 吳秉柔

B06902110 林奕萱

B06902130 蘇庭葦

### I. Introduction

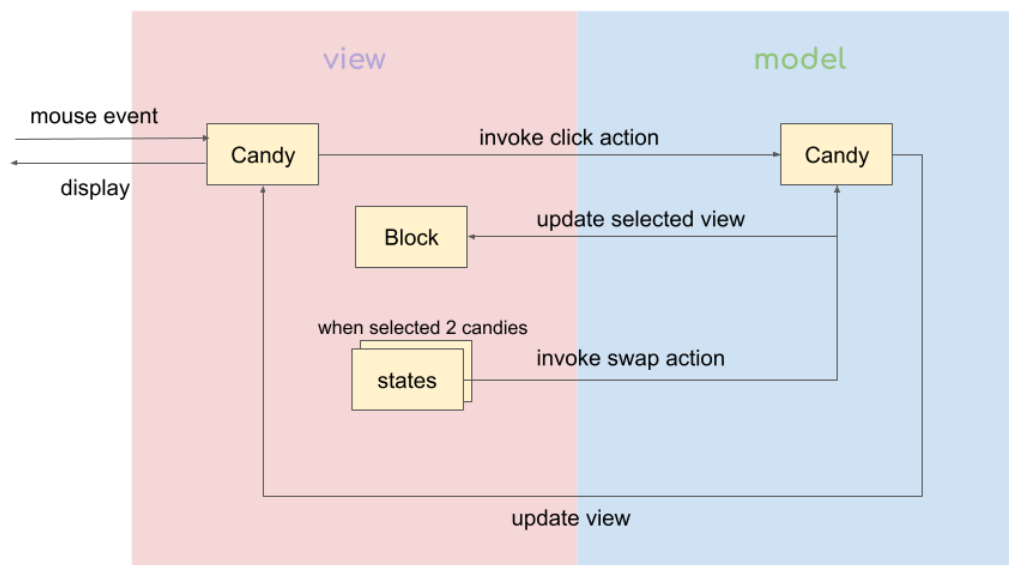
遊戲名稱: Candy Crush

遊戲介紹: 一款經典的消除遊戲, 透過滑鼠點擊交換兩顆糖果的位置符合組合規則並觸發消除效果, 搭配各式各樣的組合規則與消除效果帶來無限可能的遊戲體驗

### II. The relations between the classes of our design

#### ❖ Overview (Responsibility Analysis)

整個遊戲由model&view兩個package組成, 採取類似前後端溝通的模式。view負責呈現GUI、聆聽事件、糖果的狀態機, model負責記錄資料、規則比對、消除糖果等邏輯部分。下圖簡單呈現滑鼠點擊事件及糖果交換的發生。



#### ❖ Model

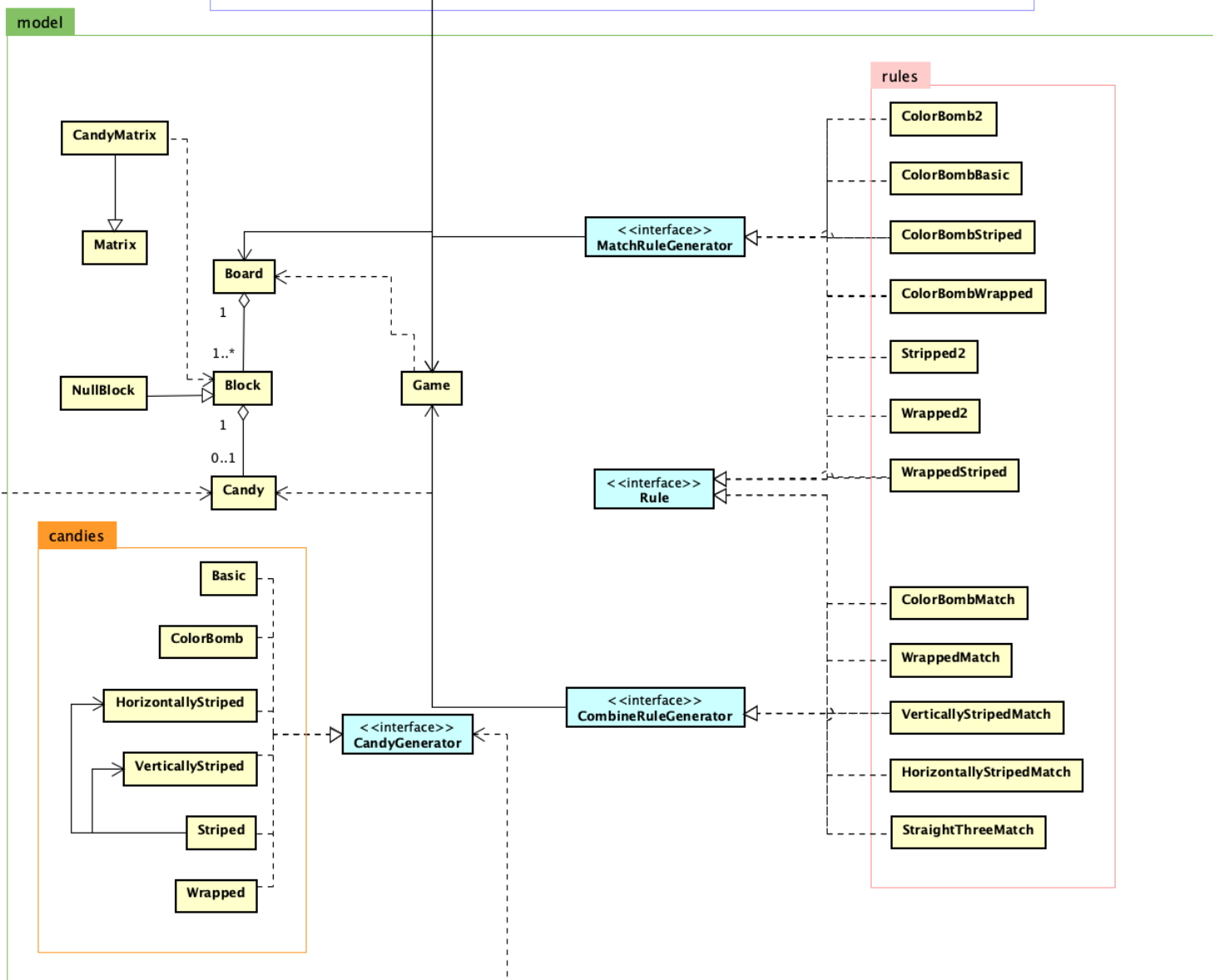
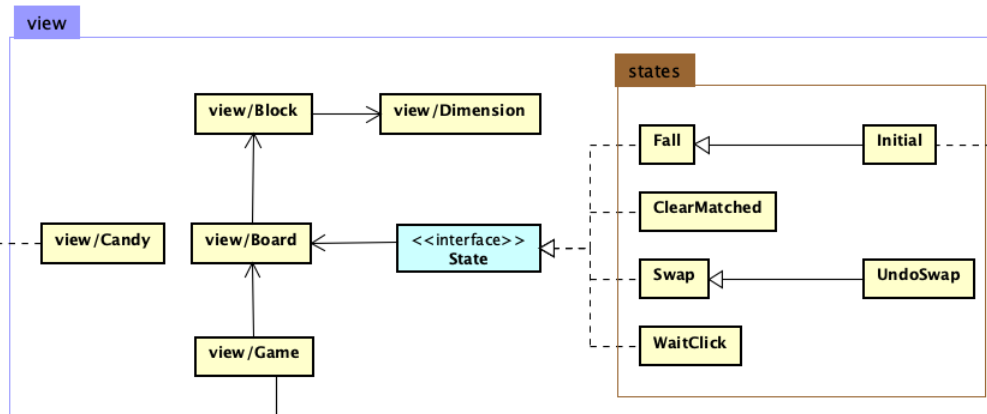
將遊戲內容分為四個層次的物件進行設計 (Game, Board, Block, Candy), 讓各層次能夠分別處理各自層級的任務。而容易有增減需求的糖果和規則, 皆具有基本內容相似但有高度不同的行為, 我們利用interface減少重複的程式內容。

- **Game**:紀錄所有消除、結合規則，以及提供消除的方法。
- **Board**:由數個Block組成，負責整個版面的控制，處理糖果落下、移動的位置、補滿空位...等。
- **Block**:用來儲存糖果及座標。
- **Candy**:定義一個糖果的基本資料，名字、外觀等。
- **rules**:包含所有消除、結合規則的細節資料。我們先定義三種interface Rule, MatchRuleGenerator, CombineRuleGenerator。如果要建立新的消除規則，需要實作Rule，設計規則細節，然後透過MatchRuleGenerator產生新的消除規則。結合規則方法相同。
- **CandyGenerator**:提供建立新糖果的介面。
- **candies**:透過CandyGenerator實作之糖果，包含各種不同類型的糖果(詳情請見 part.VI)

#### ❖ View

透過JAVA Swing、awt套件來實現GUI，處理與滑鼠的互動。另外，因為要處理等待選取、交換、取消交換、掉落等狀態，此部分設計了states來達到狀態機的功能，並透過Board來執行。

- **Board**:擴充JPanel實作Runnable執行state.run，呈現layout。
- **states**:包含Initial, WaitClick, Swap, UndoSwap, Fall, ClearMatched。當WaitClick等到選了兩個糖果，就會切換至Swap，然後Swap詢問model此次交換是否合法，不合法就會切換至UndoSwap。當ClearMatched發生時就會切換至Fall。因為Initial就是一開始都沒有糖果的時候，類似全部清空糖果要落下的時候，所以是extends Fall來實作。
- **Block**:呈現有無被滑鼠選取。
- **Candy**:擴充JLabel實作MouseListener呈現糖果外觀，並作為滑鼠點擊事件的接收者。



### III. The advantages of our design

1. 容易擴充糖果種類(顏色、功能性)
2. 容易新增新的糖果組合規則
3. 容易新增新的消除效果
4. 可以特別設定起始畫面糖果顏色分佈及種類

### IV. The disadvantages of our design

1. 消除方式侷限在相同顏色、以矩形為範圍的區塊
2. 每新增一種功能型的糖果就要就已知糖果的各種功效重新撰寫match rule
3. board目前只能是一個矩形
4. 如果場面上沒有可以消滅的糖果的話就會不能繼續進行遊戲

### V. Future & Optimization

1. Board可以是客製化的形狀

可能做法:因為目前設計是由Board來產生Block進而造成形狀, 因此要實作此部分需要更改Board中Block的產生方法。

2. 增加遊戲滿足某些條件(消除幾顆糖果、消除幾顆特定功效、特定顏色的糖果、挑戰時限內消除糖果的數量)就通關的設定、並且擴充關卡

可能做法:按照目前的設計, 需要在Game之上增加關卡及GameLoop的class, 然後將目前的Game引用其中。

3. 除了糖果之外的物品:不能被消除的障礙物、會阻擋消除的障礙物...

可能做法:因為此為與糖果同層級的概念, 但行為差異大, 因此可以基於糖果的設計增加可否消除的屬性。

4. 如果場上已經無法消除糖果會自動重新排序

可能做法:在Board的層級新增一個物件引用swap, match, clear等功能來檢視是否有潛在可消除者, 若沒有則重新一盤。

5. 增加消除音效

可能做法:引入消除的音效在rule.act()消除糖果時播放。

6. 一次只消除一組糖果(更細緻的動畫)(現在是所有符合的糖果會同時消失)

### VI. How to play your 2D game

#### ❖ 遊戲步驟:

1. 以滑鼠點擊選擇一顆糖果
2. 再選擇該糖果上下左右的糖果來交換, 如果交換後不符合消除或結合條件則會回復原狀

❖ 遊戲規則：



普通糖果



條紋糖果



包裝糖果



巧克力糖果

1. 三顆相同顏色的糖果排成一直線即可消除
2. 四顆相同顏色的糖果排成一直線即可消除並生成一顆帶條紋的糖果
3. 五顆相同顏色的糖果如果呈現L型或T型即可消除並生成一顆包裝的糖果
4. 五顆相同顏色的糖果排成一直線即可消除並生成一顆黑色巧克力糖果
5. 消除的糖果中有條紋糖果則會依照條紋的方向消去該條紋糖果的直排或橫排糖果
6. 消除的糖果中有包裝糖果則會以包裝糖果為中心消去周遭九宮格的糖果
7. 黑色巧克力糖果與普通的糖果交換位置後會把盤面上所有和普通糖果相同顏色的糖果消去
8. 兩顆條紋糖果交換後會消除並將交換中心糖果的直排與橫排的糖果們消去
9. 兩顆包裝糖果交換後會消除並以其中一顆糖果為中心消去周圍5x5的糖果們
10. 兩顆巧克力糖果交換後會消除並將整個版面的糖果都清空
11. 一顆條紋糖果和一顆包裝糖果交換後會以其中一顆為中心消去自己這一直排跟橫排還有隔壁直排跟橫排(類似十字)
12. 一顆條紋糖果和一顆黑色巧克力糖果交換後會將盤面上所有和條紋糖果同顏色的糖果轉換成條紋糖果並觸發效果
13. 一顆包裝糖果和一顆黑色巧克力糖果交換後會將盤面上所有和包裝糖果同顏色的糖果轉換成包裝糖果並觸發效果

VII. Other packages that we have used

- ❖ Java Swing JFrame, JLabel, JLayerPane, Jpanel, Joptionpane
- ❖ Java awt color, image, event

VIII. How you divide the responsibilities of the team members

- ❖ package view: 吳秉柔、林奕萱
- ❖ package model: 吳秉柔、蘇庭葦
- ❖ 報告、影片: 蘇庭葦、林奕萱