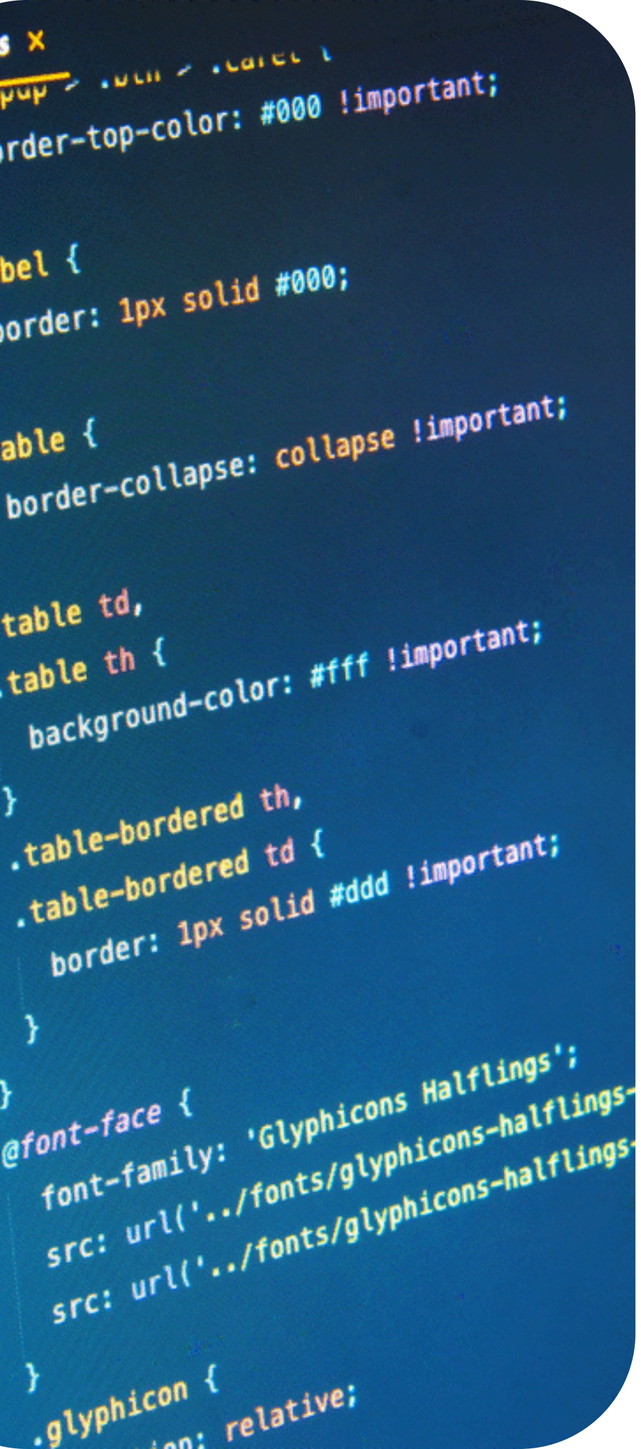
回合制打怪遊戲

112213067 陳彥哲

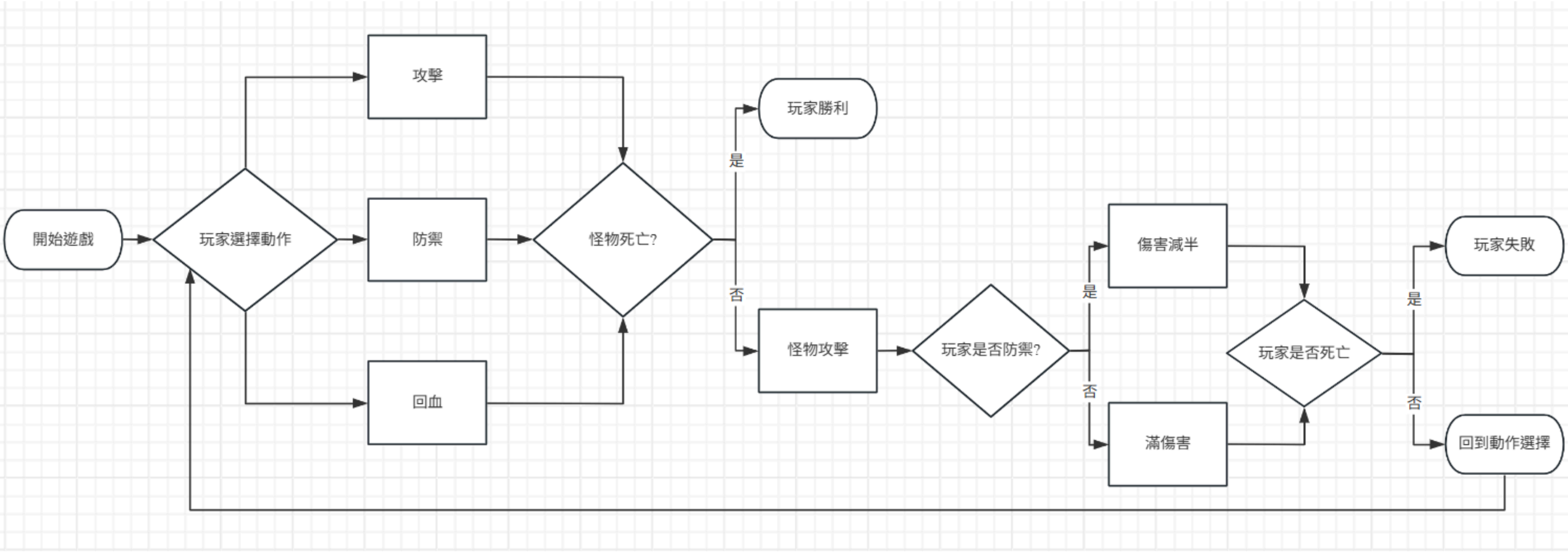


大綱

＊＊＊

- 系統流程圖
- 程式碼解析
- Demo
- 遇到的困難

流程圖


開始遊戲
玩家選擇動作
攻擊
防禦
回血
怪物死亡?
是
否
玩家勝利
怪物攻擊
玩家是否防禦?
是
否
傷害減半
滿傷害
玩家是否死亡
是
否
玩家失敗
回到動作選擇

怪物

- 怪物生命值，初始為 100
- 傳回目前生命值
- 隨機產生 8～18 的攻擊力
- 受到傷害並扣除血量（最少為 0）

```java
package com.example.demo1;

public class Monster {
    private int hp;

    public Monster() {
        this.hp = 100;
    }
    public int getHp() {
        return this.hp;
    }
    public int attack() {
        return (int)(Math.random() * 11 + 8);
    }
    public void takeDamage(int damage) {
        this.hp = Math.max(0, this.hp - damage);
    }
}
```

玩家

- 玩家生命值，初始為100
- 玩家是否正在防禦，初始為未防禦
- 傳回目前生命值&防禦狀態

```java
package com.example.demo1;

public class Player {
    private int hp;
    private boolean defending;

    public Player() {
        this.hp = 100;
        this.defending = false;
    }
    public int getHp() {
        return this.hp;
    }
    public boolean isDefending() {
        return this.defending;
    }
```

```java
public void setDefending(boolean defending) {
    this.defending = defending;
}
public int attack() {
    return (int)(Math.random() * 11 + 10);
}
public int heal() {
    int amount = (int)(Math.random() * 6 + 10);
    this.hp = Math.min(100, this.hp + amount);
    return amount;
}
public void takeDamage(int damage) {
    if (this.defending) {
        damage = damage / 2;
    }
    this.hp = Math.max(0, this.hp - damage);
}
```

設定防禦狀態

- 隨機產生10～20 的攻擊力
- 隨機回復 10～15 點生命值
- 如果正在防禦傷害值會減半
  然後從 hp 中扣除傷害，最低為 0

控制

```java
package com.example.demo1;

import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/game")
@CrossOrigin
public class BattleController {

    private Player player = new Player();
    private Monster monster = new Monster();
```

- 表示這是一個 REST API 控制器，回傳的是 JSON 而非 HTML 頁面。
- 所有路徑會以 /game 開頭，例如 /game/action
- 允許前端網頁跨網域呼叫此後端 API
- 建立玩家和怪物的實例

```java
@PostMapping("/action")
public Map<String, Object> playerAction(@RequestParam String type) {
    Map<String, Object> result = new HashMap<>();
    int playerDamage = 0, monsterDamage = 0, healAmount = 0;
    player.setDefending(false);
    switch (type) {
        case "attack":
            playerDamage = player.attack();
            monster.takeDamage(playerDamage);
            break;
        case "defend":
            player.setDefending(true);
            break;
        case "heal":
            healAmount = player.heal();
            break;
    }
```

- 當玩家透過 POST 請求，系統會根據 type 處理動作，傳回一個 JSON 結果
- 每回合預設關閉防禦
- 隨機造成 10~20 傷害，怪物扣血
- 開啟防禦，減半怪物下回合攻擊力
- 隨機回復 10~15 HP

```
    if (monster.getHp() > 0) {
        monsterDamage = monster.attack();
        player.takeDamage(monsterDamage);
    }
    result.put("playerHp", player.getHp());
    result.put("monsterHp", monster.getHp());
    result.put("playerDamage", playerDamage);
    result.put("monsterDamage", monsterDamage);
    result.put("heal", healAmount);
    result.put("defending", player.isDefending());
    if (player.getHp() <= 0) {
        result.put("status", "lost");
    } else if (monster.getHp() <= 0) {
        result.put("status", "won");
    } else {
        result.put("status", "ongoing");
    }
    return result;
}
```

- 怪物攻擊（8~18 傷害）
- 玩家扣血（若有防禦減半）
- 玩家、怪物剩餘 HP
- 玩家、怪物本回合造成的傷害
- 玩家回血值
- 玩家是否在防禦狀態

- 玩家死亡、怪物死亡、繼續遊戲

```java
@PostMapping("/reset")
public String resetGame() {
    player = new Player();
    monster = new Monster();
    return "Game reset.";
}
```

- 重設整場戰鬥，讓玩家與怪物都回到滿血狀態


DEMO

遇到的困難

- 第一次接觸spring boot
- 添加普攻、技能
- 新增怪物


謝謝大家聆聽！