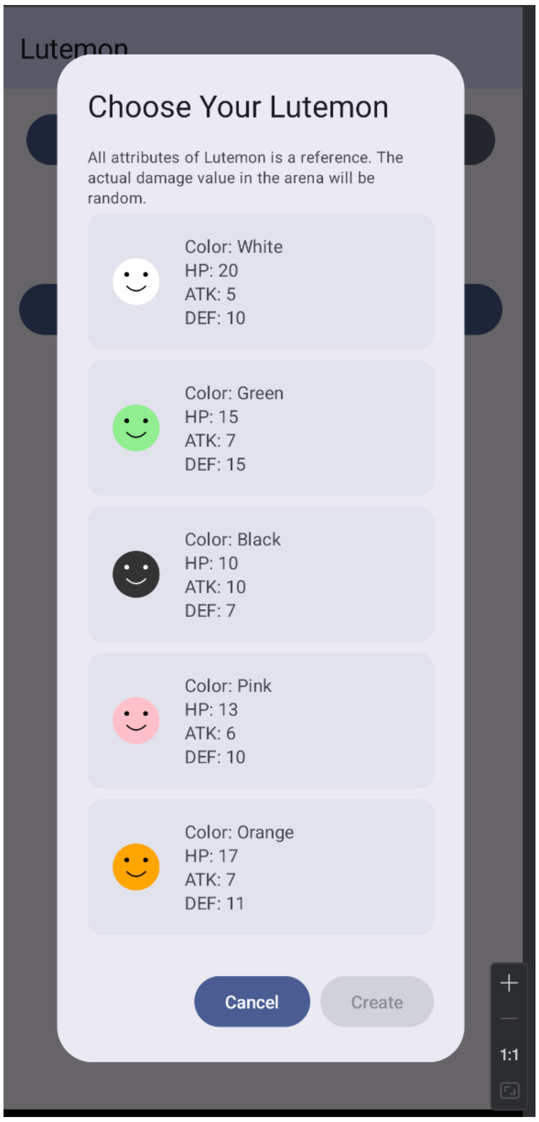
Hello, this is the documentation of the OOP course final Lutemon project.

1. **The link of my github repository where you can download my code and run it on your Android Studio:**
   https://github.com/knporter/OOP-final-Lutemon-project/tree/master
2. **The link of my video is below**, I have uploaded the video demonstration on Youtube. The video is about how to download my code from github and how to run it on Android studio and functions of my Lutemon application and how to play it. So please check the video first.
   https://www.youtube.com/watch?v=3k2Sve_IdFo
3. **Introduction of my application:**
   (1) Home Page
   There are 4 parts of my application: Home、Battle、Training、Statistic



Choose Your Lutemon

All attributes of Lutemon is a reference. The actual damage value in the arena will be random.

Color: White
HP: 20
ATK: 5
DEF: 10

Color: Green
HP: 15
ATK: 7
DEF: 15

Color: Black
HP: 10
ATK: 10
DEF: 7

Color: Pink
HP: 13
ATK: 6
DEF: 10

Color: Orange
HP: 17
ATK: 7
DEF: 11

In the Home page, we can create different Lutemon (Pink、White、Orange、Green、Black).

Different Lutemon has different attributes (HP、DEF、ATK), these different attributes is only a reference. In the battle arena, the damage will based on these different attributes.

In this part, I use OOP programming principles, for example Lutemon class.

After creating, the Lutemon is in the Home. Initially the XP points is 0. There is a RecyclerView to show different Lutemons

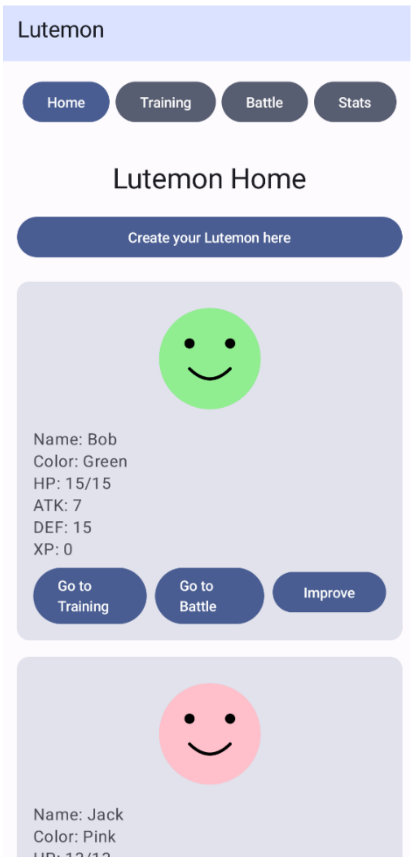
In each Lutemon card, there are 3 different buttons, Go to Training、Go to Battle、Improve.

(2) Battle page

We can move Lutemon to battle area and begin battle. The battle is a turn based. There are two different actions can be randomly chose in a turn, attack and heal. The action of each turn chosen is totally random. There are also a low possibility of Critical attack which will damage 7 health points.

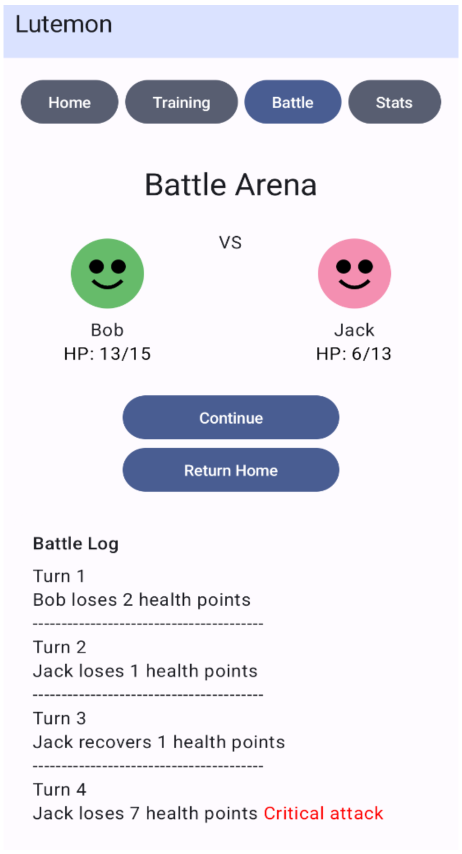
The winner of the battle will increase 1 point of XP, the loser of the battle will deduct 1, but if the XP is 0 at the beginning, XP value will not deduct.

After the battle, two Lutemon will send home automatically and the HP value will reset to full points.



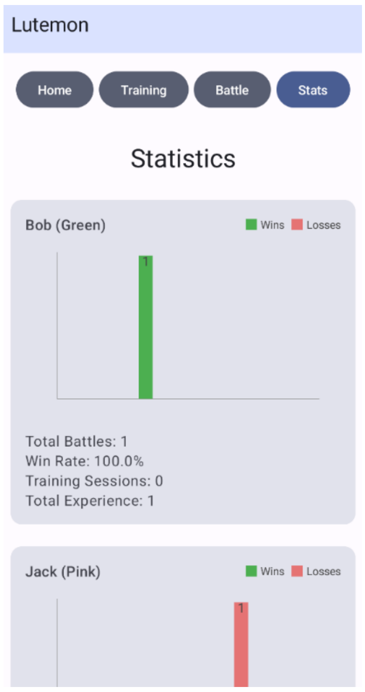
Name: Bob
Color: Green
HP: 15/15
ATK: 7
DEF: 15
XP: 0

Name: Jack
Color: Pink



```
Battle Log
Turn 1
Bob loses 2 health points
-----------------------------------------
Turn 2
Jack loses 1 health points
-----------------------------------------
Turn 3
Jack recovers 1 health points
-----------------------------------------
Turn 4
Jack loses 7 health points Critical attack
```

(3) Statistic Page



Statistics

Bob (Green)

🟩 Wins 🟥 Losses

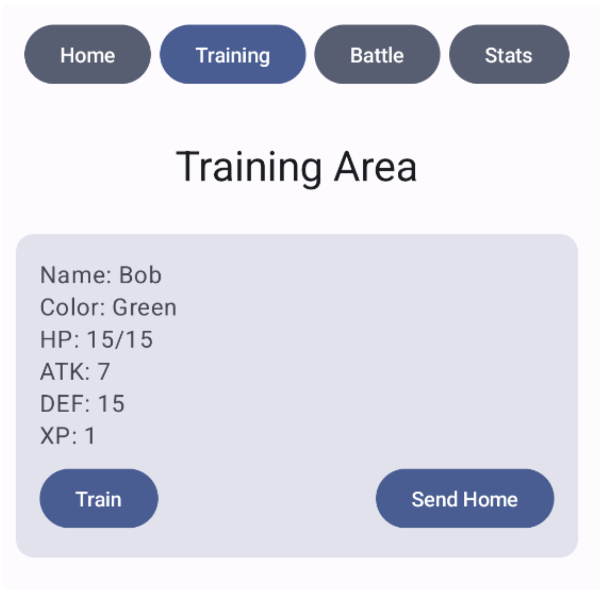
Total Battles: 1
Win Rate: 100.0%
Training Sessions: 0
Total Experience: 1

Jack (Pink)

🟩 Wins 🟥 Losses

In this page, there is a bar chart to record the battle performance of each Lutemon, total battles, winning rate and Training sessions.
The green bar is Wins and the red bar is Losses,

(4) Training page

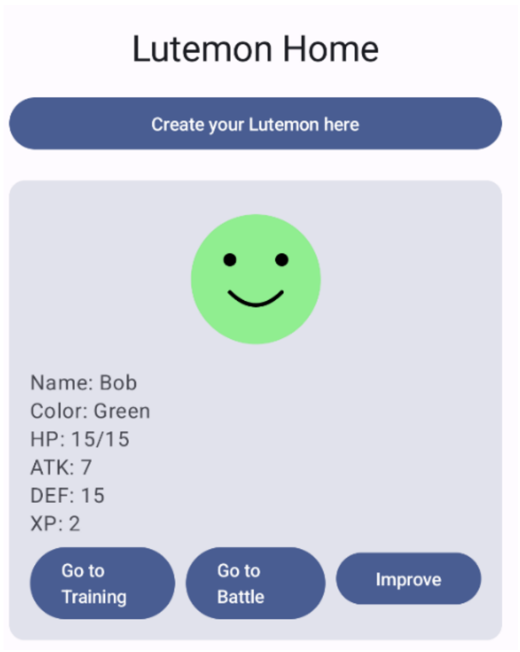


Training Area

Name: Bob
Color: Green
HP: 15/15
ATK: 7
DEF: 15
XP: 1

In this page, we can train different Lutemon, After training, the XP value will increase 1 automatically.
There is also a basic rule of Training. Each lutemon cannot train infinite times. After two times of training, it must go to the battle area and fight a duel, after that, it will gain another 2 chance of training.

(5) Improvement

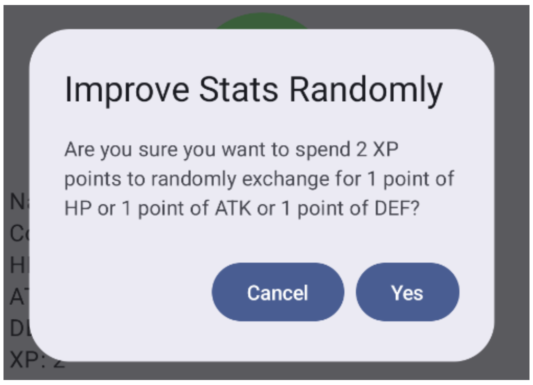


Lutemon Home

Create your Lutemon here

Name: Bob
Color: Green
HP: 15/15
ATK: 7
DEF: 15
XP: 2

In the home page, each lutemon card has an Improve button. When you click it, it will open a window shows that ‘Are you sure you want to spend 2 XP to randomly exchange one point of HP or ATK or DEF?’. It means that the increase outcome is totally random.

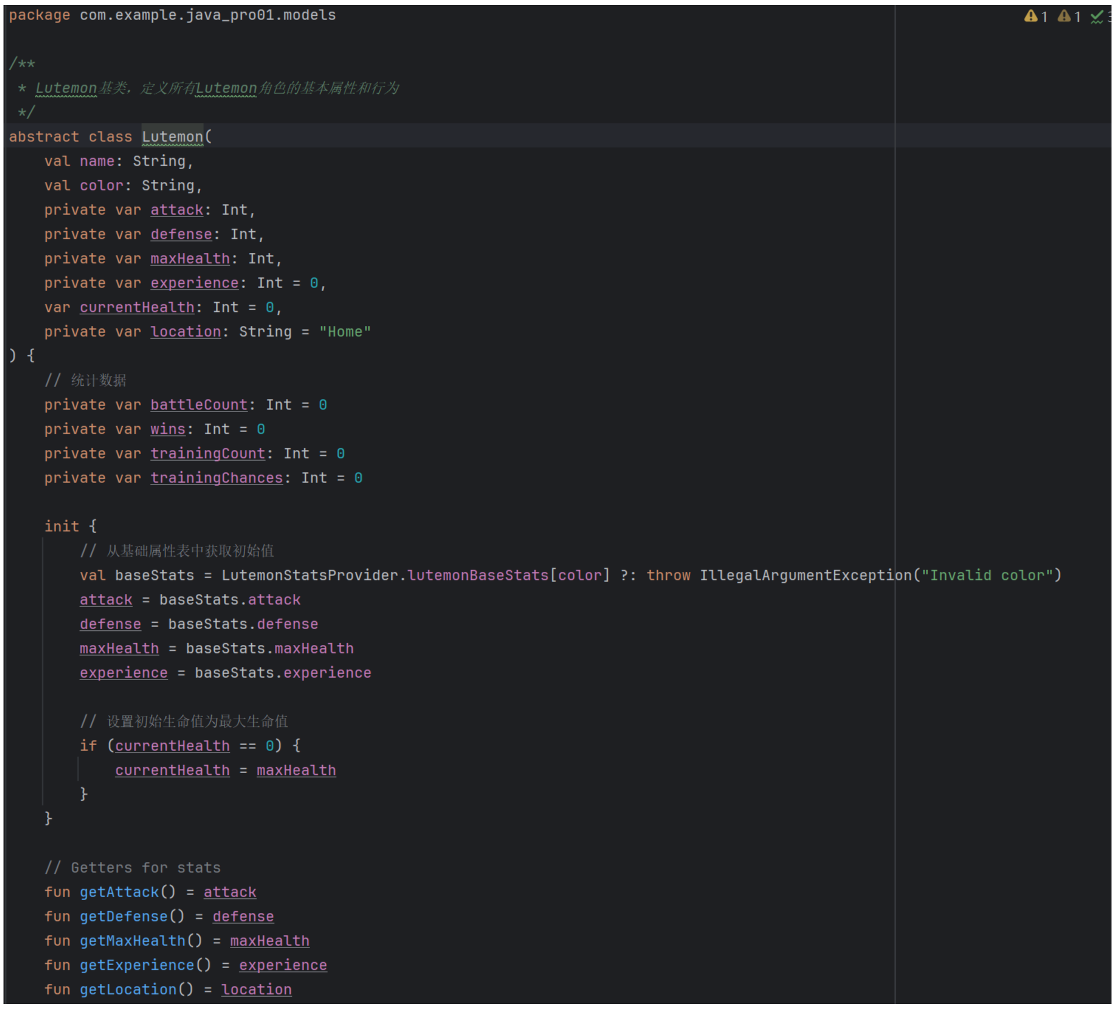


Improve Stats Randomly

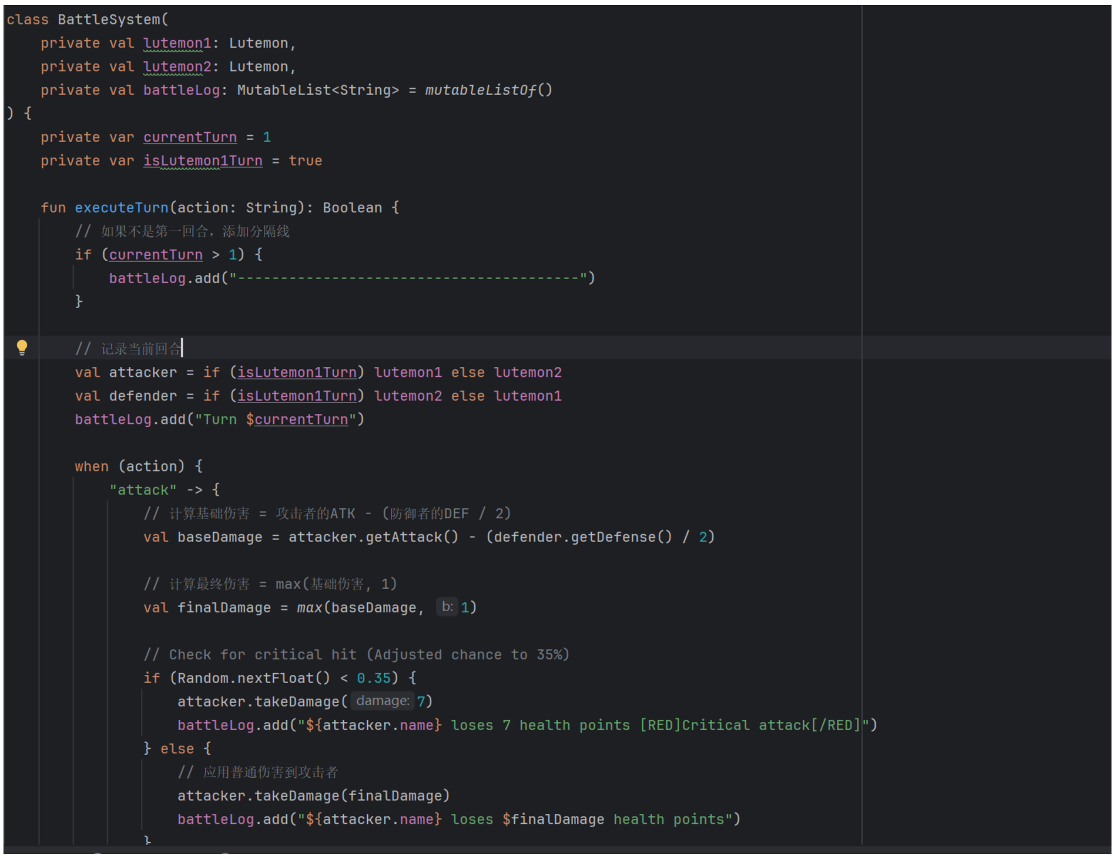
Are you sure you want to spend 2 XP points to randomly exchange for 1 point of HP or 1 point of ATK or 1 point of DEF?

(6) Code part

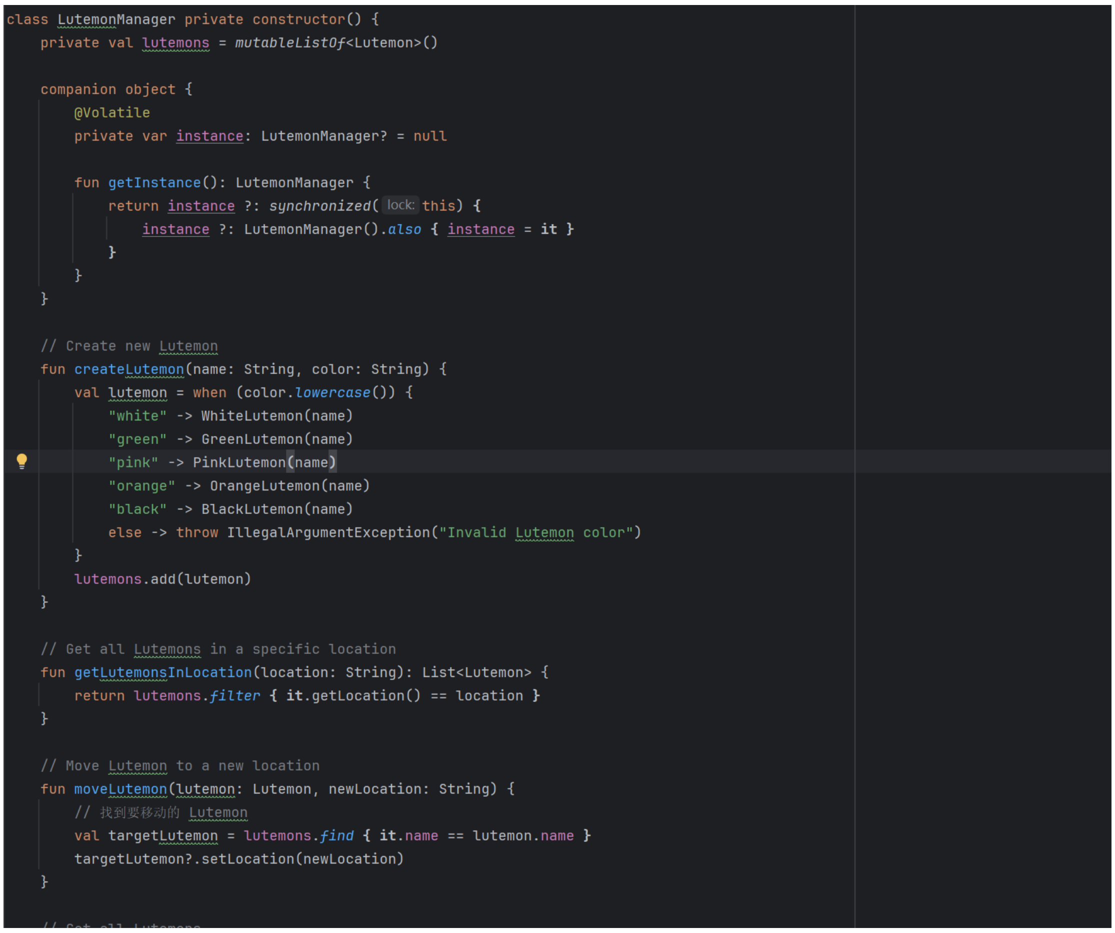
We have different class (BattleSystem、LutemonManager、Lutemon). Each class has different attributes and functions, which follows strictly of the OOP programming principle.



```kotlin
package com.example.java_pro01.models

/**
 * Lutemon基类，定义所有Lutemon角色的基本属性和行为
 */
abstract class Lutemon(
    val name: String,
    val color: String,
    private var attack: Int,
    private var defense: Int,
    private var maxHealth: Int,
    private var experience: Int = 0,
    var currentHealth: Int = 0,
    private var location: String = "Home"
) {
    // 统计数据
    private var battleCount: Int = 0
    private var wins: Int = 0
    private var trainingCount: Int = 0
    private var trainingChances: Int = 0

    init {
        // 从基础属性表中获取初始值
        val baseStats = LutemonStatsProvider.lutemonBaseStats[color] ?: throw IllegalArgumentException("Invalid color")
        attack = baseStats.attack
        defense = baseStats.defense
        maxHealth = baseStats.maxHealth
        experience = baseStats.experience

        // 设置初始生命值为最大生命值
        if (currentHealth == 0) {
            currentHealth = maxHealth
        }
    }

    // Getters for stats
    fun getAttack() = attack
    fun getDefense() = defense
    fun getMaxHealth() = maxHealth
    fun getExperience() = experience
    fun getLocation() = location
```



```kotlin
class BattleSystem(
    private val lutemon1: Lutemon,
    private val lutemon2: Lutemon,
    private val battleLog: MutableList<String> = mutableListOf()
) {
    private var currentTurn = 1
    private var isLutemon1Turn = true

    fun executeTurn(action: String): Boolean {
        // 如果不是第一回合，添加分隔线
        if (currentTurn > 1) {
            battleLog.add("-----------------------------------------")
        }

        // 记录当前回合
        val attacker = if (isLutemon1Turn) lutemon1 else lutemon2
        val defender = if (isLutemon1Turn) lutemon2 else lutemon1
        battleLog.add("Turn $currentTurn")

        when (action) {
            "attack" -> {
                // 计算基础伤害 = 攻击者的ATK - (防御者的DEF / 2)
                val baseDamage = attacker.getAttack() - (defender.getDefense() / 2)

                // 计算最终伤害 = max(基础伤害, 1)
                val finalDamage = max(baseDamage, b: 1)

                // Check for critical hit (Adjusted chance to 35%)
                if (Random.nextFloat() < 0.35) {
                    attacker.takeDamage( damage: 7)
                    battleLog.add("${attacker.name} loses 7 health points [RED]Critical attack[/RED]")
                } else {
                    // 应用普通伤害到攻击者
                    attacker.takeDamage(finalDamage)
                    battleLog.add("${attacker.name} loses $finalDamage health points")
                }
```



```kotlin
class LutemonManager private constructor() {
    private val lutemons = mutableListOf<Lutemon>()

    companion object {
        @Volatile
        private var instance: LutemonManager? = null

        fun getInstance(): LutemonManager {
            return instance ?: synchronized( lock: this) {
                instance ?: LutemonManager().also { instance = it }
            }
        }
    }

    // Create new Lutemon
    fun createLutemon(name: String, color: String) {
        val lutemon = when (color.lowercase()) {
            "white" -> WhiteLutemon(name)
            "green" -> GreenLutemon(name)
            "pink" -> PinkLutemon(name)
            "orange" -> OrangeLutemon(name)
            "black" -> BlackLutemon(name)
            else -> throw IllegalArgumentException("Invalid Lutemon color")
        }
        lutemons.add(lutemon)
    }

    // Get all Lutemons in a specific location
    fun getLutemonsInLocation(location: String): List<Lutemon> {
        return lutemons.filter { it.getLocation() == location }
    }

    // Move Lutemon to a new location
    fun moveLutemon(lutemon: Lutemon, newLocation: String) {
        // 找到要移动的 Lutemon
        val targetLutemon = lutemons.find { it.name == lutemon.name }
        targetLutemon?.setLocation(newLocation)
    }

    // Get all Lutemons
```

(7) Features of my project

In addition to the basic requirements, I have complete almost all Bonus Features

(a) RecyclerView: In the home page, You can scroll through the created Lutemon

(b) Lutemon have images: In the Lutemon list, different color Lutemon have different smile face image.

(c) Battle Visualization: In the battle area, one Lutemon move to another one, it means attack another Lutemon, If one Lutemon jump up, it means heal itself.

(d) Turn-based combat: In the battle area, the battle is totally turn-based. You only need to click the battle button once and then it will battle automatically on a turn based

(e) Statistics: In the statistic page, we have recorded the battle data of different Lutemon, including winning rate、total battles etc…

(f) No Death: Instead of dying, the defeated Lutemon will send back to home and reset its HP value to the initial status.

(g) Randomness in Battles: During the battle, there are two different actions in one turn, attack and heal. Which action will perform is totally random, decide by the system, there will also some probabilities of critical attack (15%-20%) which will damage 7 health points.

(h) Fragments: In the training area, there will be a progress bar showing the training progress.

(i) Statistics visualization: In the statistic page, there is bar chart to record each Lutemon's performance.

(8) The end

That's all of my Lutemon project. Thanks for your checking.