移动软件开发课程作业（计科23-123）

1. 待办事项清单 (To-Do List)

- **项目描述：** 一个经典的入门项目。用户可以查看待办事项列表，添加新事项，并标记已完成或删除事项。
- **需掌握的核心技能：**
  - **列表视图：** 使用 `RecyclerView` 高效地展示任务列表。
  - **数据适配器：** 编写 `Adapter` 将数据（任务列表）绑定到 `RecyclerView` 上。
  - **`Intent` 应用：**
    i. 主界面（`MainActivity`）显示列表。点击“添加”按钮，使用 `Intent` 启动 `AddTodoActivity`。
    ii. 在 `AddTodoActivity` 中输入新任务后，将任务内容通过 `Intent` 返回给 `MainActivity` 并更新列表。
    iii. 点击列表中的某一项，可以启动一个 `DetailActivity` 来查看或编辑该项任务。
- **扩展方向：** 使用 Room 数据库或 `SQLite` 替代简单的 `ArrayList` 来持久化存储数据。

2. 趣味问答App (Quiz App)

- **项目描述：** 应用内置一套选择题。用户开始答题，应用判断对错并计分，最后展示答题结果。
- **需掌握的核心技能：**
  - **UI组件：** 使用 `TextView` 显示问题，`RadioButton` 或 `Button` 作为选项。
  - **数据管理：** 在代码中用一个类或列表来管理所有问题和答案。
  - **`Intent` 应用：**
    i. `WelcomeActivity`：包含应用介绍和“开始答题”按钮。
    ii. 点击按钮后，用 `Intent` 启动 `QuizActivity` 开始答题。
    iii. 所有题目回答完毕后，`QuizActivity` 启动 `ResultActivity`，并通过 `Intent` 将最终得分传递过去进行展示。
- **扩展方向：** 从本地文件或网络API获取题目，增加错题本功能。

3. 猜数字游戏 (Guess the Number)

- **项目描述：** 程序随机生成一个1到100之间的数字，玩家输入猜测的数字，程序给出“偏大”、“偏小”或“猜对了”的提示，直到玩家猜中为止。
- **需掌握的核心技能：**
  - **随机数生成：** `Random` 类的使用。
  - **输入与反馈：** 获取 `EditText` 的输入，并通过 `TextView` 或 `Toast` 给出提示。
  - **游戏逻辑：** 循环判断逻辑，记录猜测次数。
  - **`Intent` 应用：** 游戏结束后，可以跳转到一个 `ResultActivity`，通过 `Intent` 传递玩家的“所用次数”，并显示“恭喜你，用了X次猜中！”。
- **扩展方向：** 增加游戏难度选择（如数字范围），或添加一个排行榜。

4. 记忆卡片配对游戏 (Memory Card Matching Game)

- **项目描述：** 类似于“翻牌记忆”游戏。界面上有多组成对的、背面向上的卡片。玩家每次翻开两张，如果图案相同则消除，不同则翻回去。目标是消除所有卡片。
- **需掌握的核心技能：**
  - **游戏面板：** 使用 `GridView` 或 `GridLayout` 动态生成卡片面板。
  - **游戏状态管理：** 需要变量来记录当前翻开的卡片、已匹配的卡片对数等。
  - **定时器：** 使用 `Handler` 或 `Timer` 实现“翻错后延迟一秒再翻回去”的效果。
  - **`Intent` 应用：** 游戏胜利后，启动 `WinActivity`，通过 `Intent` 传递游戏用时或得分。
- **扩展方向：** 增加不同的难度等级（如 4x4, 6x6 面板），加入计分和计时功能。

5. 摇骰子应用 (Dice Roller)

- **项目描述：** 一个非常简单有趣的应用。界面上有一个或多个骰子，用户点击按钮或摇动手机，骰子会随机滚动并显示新的点数。
- **需掌握的核心技能：**
  - **`ImageView` 更新：** 准备6张骰子点数的图片，根据随机数结果，使用 `setImageResource()` 来更新 `ImageView` 显示的图片。
  - **动画效果：** 可以为 `ImageView` 添加一个简单的旋转或晃动动画，让摇骰子过程更逼真。
  - **传感器（可选）：** 监听加速度传感器，实现“摇一摇”手机来掷骰子的功能。
  - **`Intent` 应用：** 添加一个“设置”菜单项，点击后使用 `Intent` 启动 `SettingsActivit`

`y`，在设置页中，用户可以选择要显示的骰子数量（1个或2个），然后将设置结果返回给主界面。

- **扩展方向：** 允许多个骰子同时滚动，并计算总点数。

6. 打地鼠 (Whac-A-Mole)

- **项目描述：** 一款考验反应速度的快节奏游戏。地鼠从洞中随机钻出，玩家需要在限定时间内尽可能多地击中它们，非常解压和有趣。
- **需掌握的核心技能：**
  - **定时与延迟任务：** 这是本项目的核心。学习使用 `Handler` 的 `postDelayed()` 方法或 `CountDownTimer` 来控制地鼠的出现和消失，以及游戏倒计时。
  - **动态UI更新：** 在Java代码中，根据随机逻辑动态地改变地鼠 `ImageView` 的可见性（`setVisibility`）或图片资源（`setImageResource`）。
  - **布局管理：** 使用 `GridLayout` 或嵌套的 `LinearLayout` 来创建一个 3x3 或 4x4 的地鼠洞网格。
  - **事件处理：** 响应玩家对地鼠的点击事件，并更新分数。
  - **游戏循环：** 理解并实现一个基本的游戏主循环（倒计时、随机出现地鼠、结束判断）。
- **扩展方向：**
  - 随机出现不同类型的“地鼠”，比如打中普通地鼠+10分，打中戴安全帽的地鼠-5分。
  - 增加游戏难度选择，比如地鼠出现的速度越来越快。

7. 老虎机 (Slot Machine)

- **项目描述：** 模拟经典的老虎机游戏，点击按钮后，三个或更多的卷轴开始“滚动”，并最终随机停止。如果图案组合符合中奖规则，则玩家获胜。这种期待感和随机性本身就很有趣。
- **需掌握的核心技能：**
  - **动画实现：** 这是本项目的亮点。学生可以学习使用简单的 `Frame Animation`（帧动画）或 `TranslateAnimation`（位移动画）来模拟卷轴滚动的视觉效果。
  - **资源数组：** 将所有卷轴上可能出现的图案资源ID（例如樱桃、铃铛、数字7等）存储在一个整型数组中，便于随机选取。
  - **组合逻辑判断：** 当滚动停止后，编写逻辑来判断三个 `ImageView` 的图案是否构成了中奖组合（例如，“三个7”或“三个樱桃”）。
  - **用户反馈：** 根据是否中奖，通过 `Toast`、`TextView` 或弹窗给用户明确的反馈。

- **扩展方向：**
  - 引入“积分”或“金币”系统，玩家有初始金币，每次摇动消耗金币，中奖则获得金币。
  - 实现一个更逼真的滚动效果，让每个卷轴滚动的速度和停止时间有细微差别。

8. 井字棋 (Tic-Tac-Toe)

- **项目描述：** 永恒的经典双人对战游戏。学生可以实现“玩家 vs 玩家”模式，或者挑战自己，编写一个简单的“玩家 vs 电脑” AI。规则简单，但逻辑严谨，能带来纯粹的策略乐趣。
- **需掌握的核心技能：**
  - **游戏状态管理：** 使用一个二维数组 (如 `int[3][3]`) 来存储棋盘的状态（空、X、或 O），这是数据驱动UI的绝佳实践。
  - **算法逻辑：** 编写核心的游戏逻辑，包括：轮流下棋、判断每次落子后是否有玩家获胜（检查行、列、对角线）、以及判断是否平局。
  - **UI与数据同步：** 玩家在界面上点击一个格子，程序更新二维数组，并根据数组的新状态刷新UI（显示X或O）。
  - **重置功能：** 实现一个“再玩一局”的按钮，用于清空棋盘数据和UI，重置游戏状态。
- **扩展方向：**
  - **实现简单AI：** 编写一个电脑对手。最简单的AI可以是随机在空格子下棋；进阶一点的AI可以懂得在自己快要赢的时候下在关键位置，或者在玩家快要赢的时候进行防守。
  - **添加棋盘线和获胜动画：** 获胜时，可以在棋盘上画一条线穿过三个连在一起的棋子。

9. “西蒙说”记忆游戏 (Simon Says - Memory Game)

- **项目描述：** 一款考验短期记忆力的经典电子游戏。游戏会随机生成一个越来越长的颜色（或声音）序列，玩家需要准确地复现这个序列。节奏逐渐加快，非常考验专注力。
- **需掌握的核心技能：**
  - **事件序列处理：** 使用 `ArrayList` 或队列来存储和管理游戏生成的指令序列（比如，`[红, 蓝, 蓝, 黄]`）。
  - **定时器与序列播放：** 使用 `Handler` 和 `postDelayed` 来按顺序、有间隔地“播放”指令序列给玩家看（例如，让红色按钮高亮1秒，然后恢复，0.5秒后再让蓝色按钮高亮）。
  - **声音反馈：** 学习使用 `SoundPool` 类。在播放序列或玩家点击按钮时，播放对应的短促音效，极大地提升游戏体验。
  - **用户输入与比较：** 捕获玩家的点击序列，并与游戏生成的序列进行逐一比较，以判断对错。

- **扩展方向：**
  - **增加难度等级：** 不同的难度对应不同的序列播放速度或序列类型。
  - **排行榜：** 使用 `SharedPreferences` 来记录玩家达到的最高关卡数，并显示在游戏主界面。

10. 自由发挥