FanFanLokMapper Project Description

Overview

FanFanLokMapper is an Android application designed to automatically detect and locate memory matching card positions from static screenshot images. The application uses computer vision techniques to identify rectangular cards arranged in a grid layout and provides coordinate data for external automation programs.

Target Platform

- **OS**: Android 15
- **Architecture**: Personal use application (not for public distribution)
- **Framework**: Jetpack Compose with Kotlin

Core Functionality

Image Processing Pipeline

1. **Image Input**: Manual selection via file picker supporting PNG and JPG formats
2. **Border Detection**: Identifies rectangular cards using distinct colored border recognition
3. **Size Filtering**: Applies minimum/maximum size thresholds to filter out noise and false positives
4. **Grid Analysis**: Fixed 4 rows × 6 columns layout
5. **Position Mapping**: Calculates center point coordinates for each detected card

Output Generation

- **Visual Overlay**: Green rectangle indicators displayed over detected card positions
- **Interactive Overlay**: Long press on rectangle overlays to remove them manually
- **JSON Export**: Simple coordinate data for external program integration
- **Debug Console**: Detailed logging for troubleshooting and verification

Technical Specifications

Input Requirements

- **File Formats**: PNG, JPG
- **Card Layout**: Fixed 4 rows × 6 columns grid (24 cards total)
- **Card Characteristics**: Rectangular shape with distinct colored borders
- **Image Source**: Static screenshots from memory matching games

Output Format

```json
{
  "cardPositions": [
    {
      "id": 0,
      "centerX": 150,
      "centerY": 200
    },
    {
      "id": 1,
      "centerX": 250,
      "centerY": 200
    }
  ]
}
```

Error Handling

- **Detection Failures**: Reports issues via console logging without fallback modes
- **Invalid Grid Detection**: Logs error details and confidence scores
- **File Format Errors**: User-friendly error messages for unsupported formats

Use Case Workflow

1. User selects screenshot image through file picker
2. Application processes image to detect card borders in a 4×6 grid layout
3. Green overlay rectangles appear over detected card positions
4. User can long press on overlay rectangles to remove incorrect detections
5. Coordinate data exported to JSON format with simple pixel coordinates
6. Debug information logged to console
7. External automation program consumes JSON coordinate data

Development Considerations

- **Performance**: Optimized for single-image processing (not real-time)
- **Accuracy**: Border detection algorithm tuned for 4×6 card grid layouts with size threshold filtering
- **Noise Filtering**: Minimum/maximum size thresholds to eliminate false positives and detection noise
- **User Correction**: Manual overlay removal via long press for incorrect detections
- **Coordinate Simplicity**: Center point coordinates provide sufficient accuracy for automation clicking
- **Extensibility**: Modular design for future enhancement of detection algorithms
- **Debugging**: Comprehensive logging for detection accuracy verification

Integration Points

- **External Programs**: JSON coordinate export for automation software
- **File System**: Local storage for processed images and output data
- **User Interface**: Simple, functional design focused on processing efficiency

This application serves as a specialized tool for extracting spatial data from memory card game screenshots, enabling automated interaction with such games through precise coordinate mapping.