🛠️ Phase 1: Card Detection Prototype

Objective: Detect and identify individual cards on the table.

Steps:

1. **Collect Card Images**
   - Capture high-quality images of each Baloot card from various angles and lighting conditions.
   - Label them using a tool like LabelImg.
2. **Train a Card Detection Model**
   - Use YOLOv8 (recommended for speed and accuracy).
   - Classify cards by **rank and suit**.
   - Test on sample images to verify correct detection.
3. **Build a Basic Interface**
   - Use OpenCV to read from a webcam.
   - Show bounding boxes and detected card labels in real time.

🧑‍🤝‍🧑 Phase 2: Player Association

Objective: Determine which player played which card.

Steps:

1. **Define Player Zones**
   - Use a top-down or angled camera view.
   - Divide the screen into 4 areas (one per player).
   - When a card is detected in a zone, assign it to that player.
2. **Track Card Plays**
   - Log the sequence of plays and card positions.
   - Label which card belongs to which player in each round.

🧠 Phase 3: Game Logic Implementation (Baloot Rules)

Objective: Apply صن and حكم rules to determine trick winners and assign points.

Steps:

1. **Build a Game Engine**
   - Write Python functions to determine:
     - Trump suit
     - Trick winner
     - Team ownership of each card
     - Bonus rules (Baloot, Ashkal, etc.)

2. **Integrate With Detected Data**
    - Pass detected card and player info to the game engine.
    - Update team scores after each trick.

📊 Phase 4: Score Tracking and UI

Objective: Display team scores and game progress in real time.

Steps:

1. **Develop a Dashboard**
    - Use **Tkinter**, **Streamlit**, or a simple HTML page.
    - Show:
        - Current scores
        - Round winner
        - Cards played
2. **Enable Game History**
    - Log all rounds in a CSV or database.
    - Optionally provide replay/export options.

🎁 Bonus Phase: Advanced Features

Optional Enhancements:

- Voice or text announcements of trick winners.
- Detect bidding phase automatically (صن vs حكم).
- Add hand/gesture tracking (e.g., to trigger card detection).
- Train the model to detect overlapping or rotated cards.

✅ Roadmap

- **Card detection with YOLOv8**
- Player card zone mapping
- صن/حكم game engine
- Real-time scoreboard UI
- Game history and export