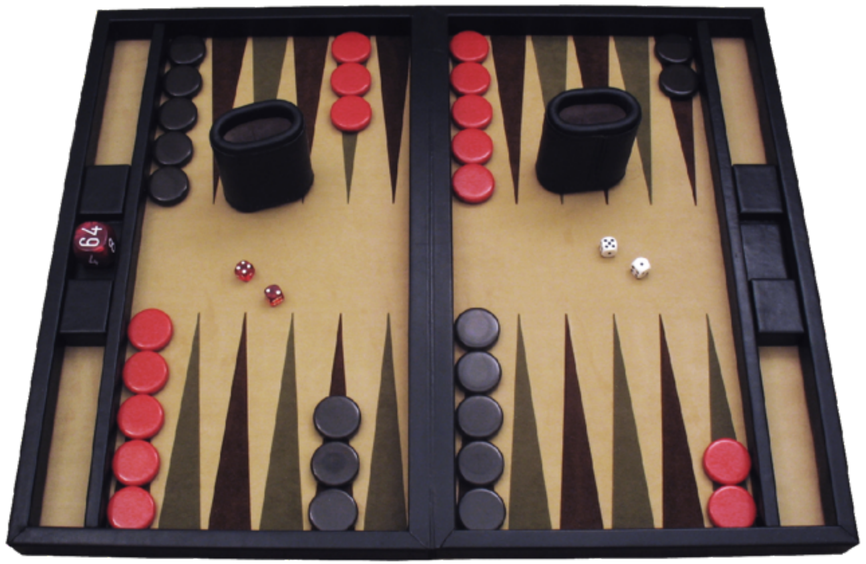
# Backgammon 4 Real

**Play Backgammon against the computer on a real board**



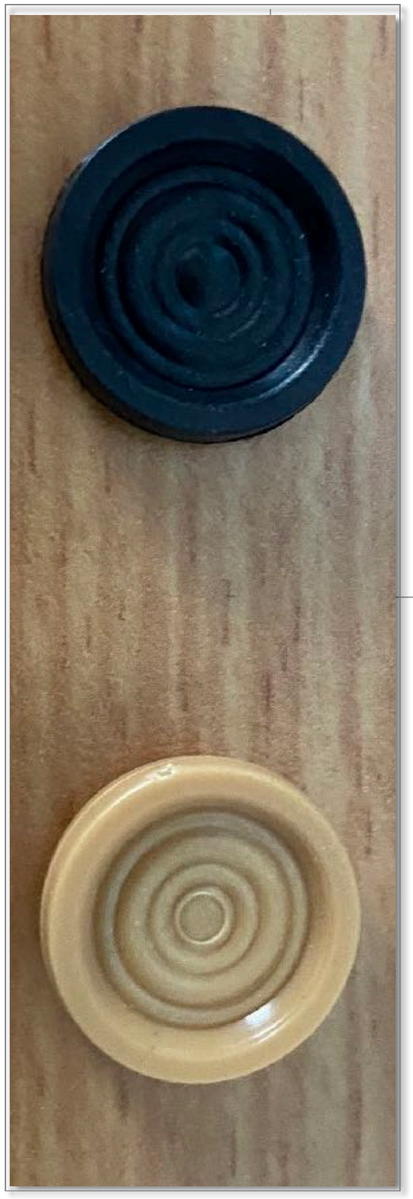
**Oz Mishli & Or Hayat**

Introduction to Computational and Biological Vision 2021

Department of Computer Science, Ben-Gurion University

# Project Goals

- Play Backgammon vs. an AI-powered computer rival
- Using a real Backgammon board and checkers
- Very good user experience
- A system that would be easy to set up, with standard components

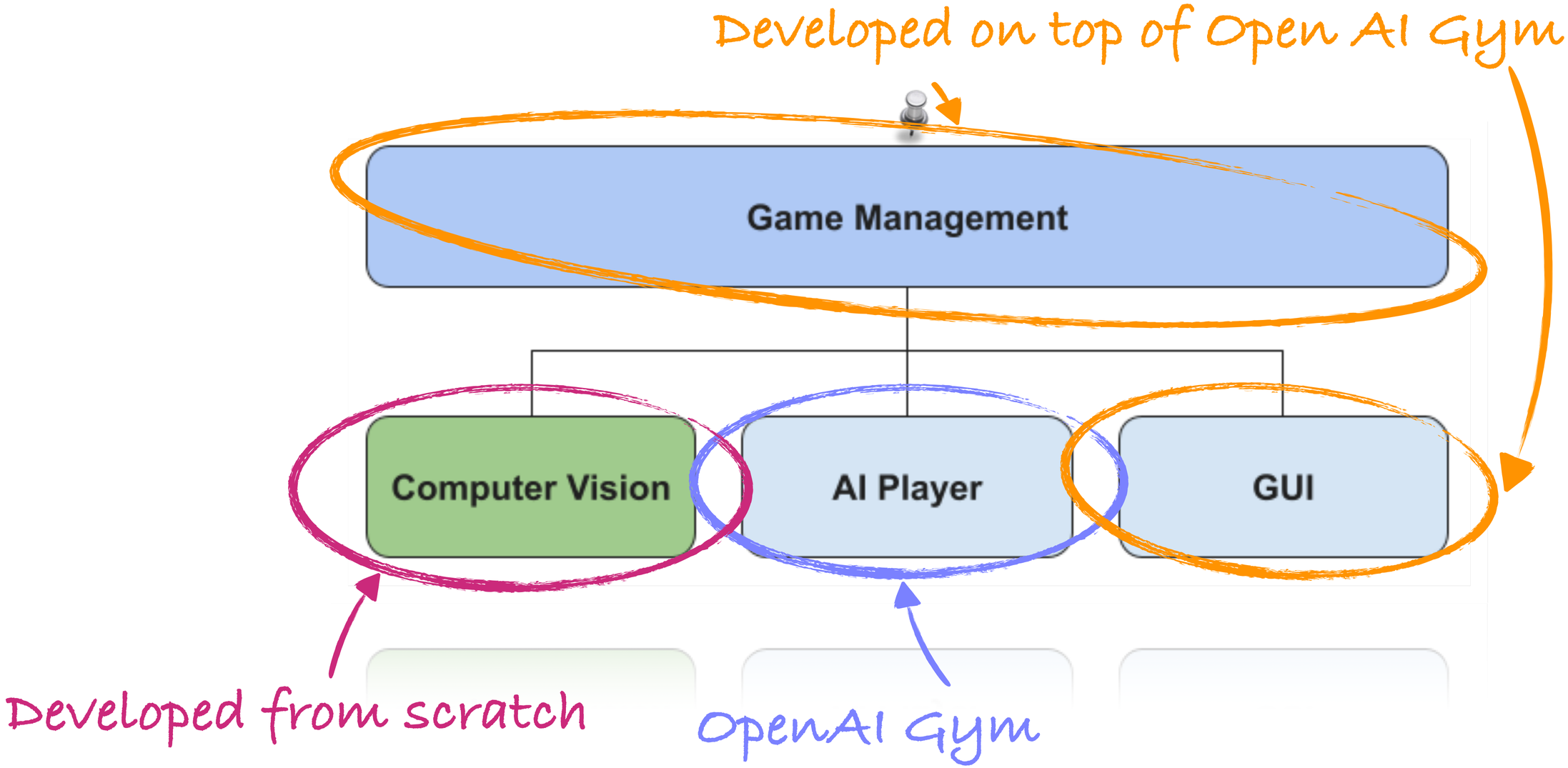


Huge variety of colors and board types

# Design Principles

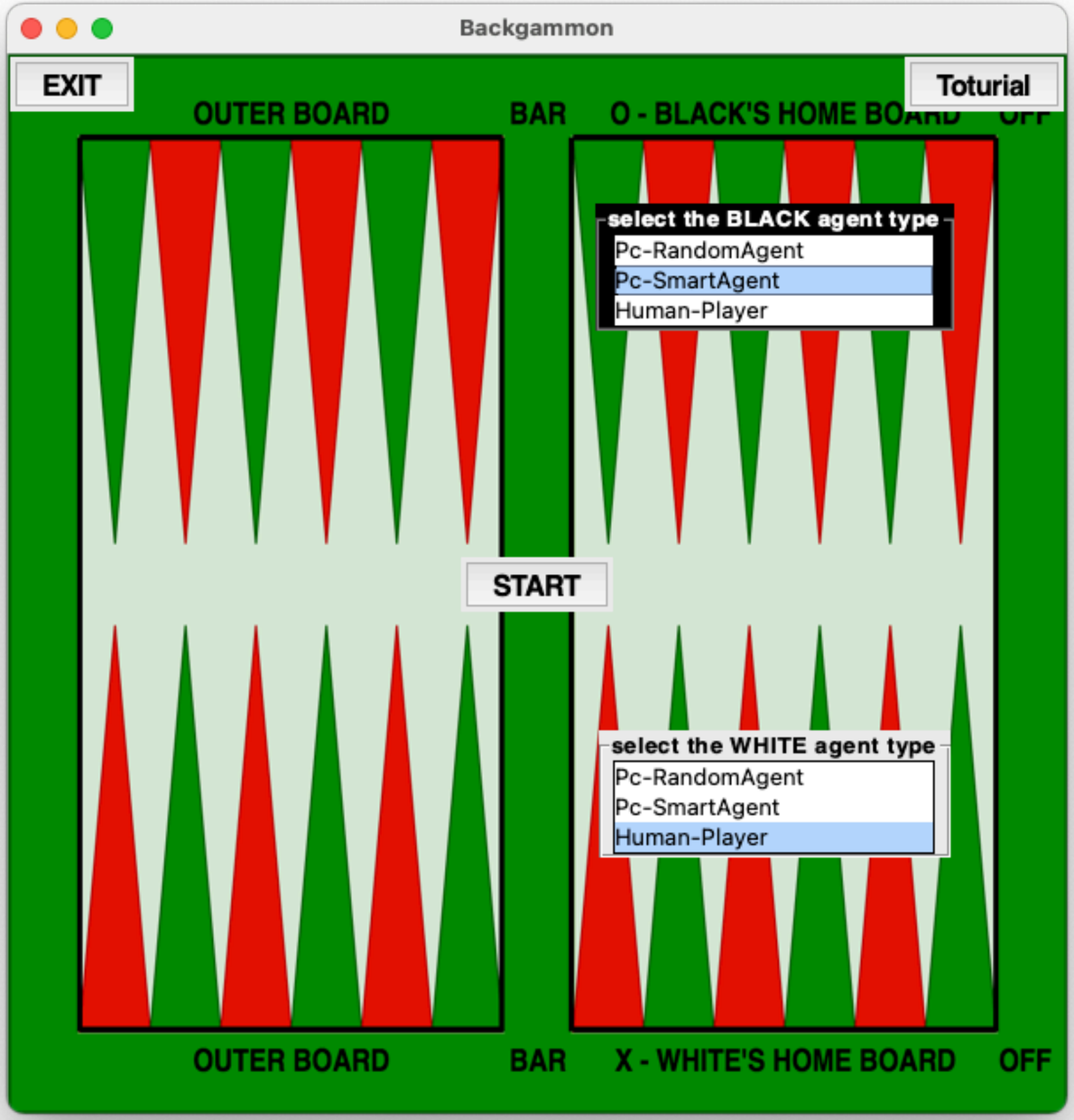
- Standard laptop and USB-webcam
- Backgammon board on the table
- Board must be fully acquired by the image
- Automated and seamless board/camera calibration
- Smooth user experience

# System Design



# GUI & AI Player

- Built using Python Tkinter library
- Multiple adaptations done for the project
- The game takes place on the real board, captured during the game
- Player is moving both checkers (human and PC)
- PC moves are broken to one by one: easier to follow by the player
- Dice roll is virtual for both players — for smoothness
- Semi automated: need to confirm after move



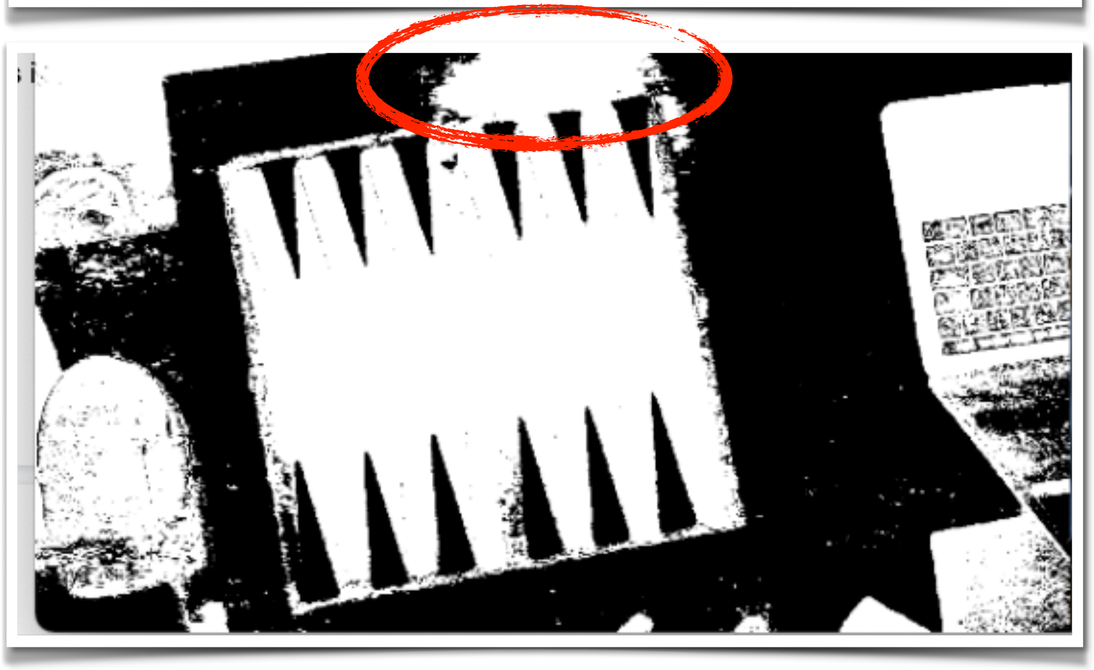
# Computer Vision Module: Challenges

- Checkers cover important visual features of the board during the game (e.g. triangles)
- Many small checkers (30), freely moving with no fixed locations on the board
- The board is not smooth, its color is not distinctive
- Black checkers color is identical to the black triangles

Chosen a "stateful board capturing" approach

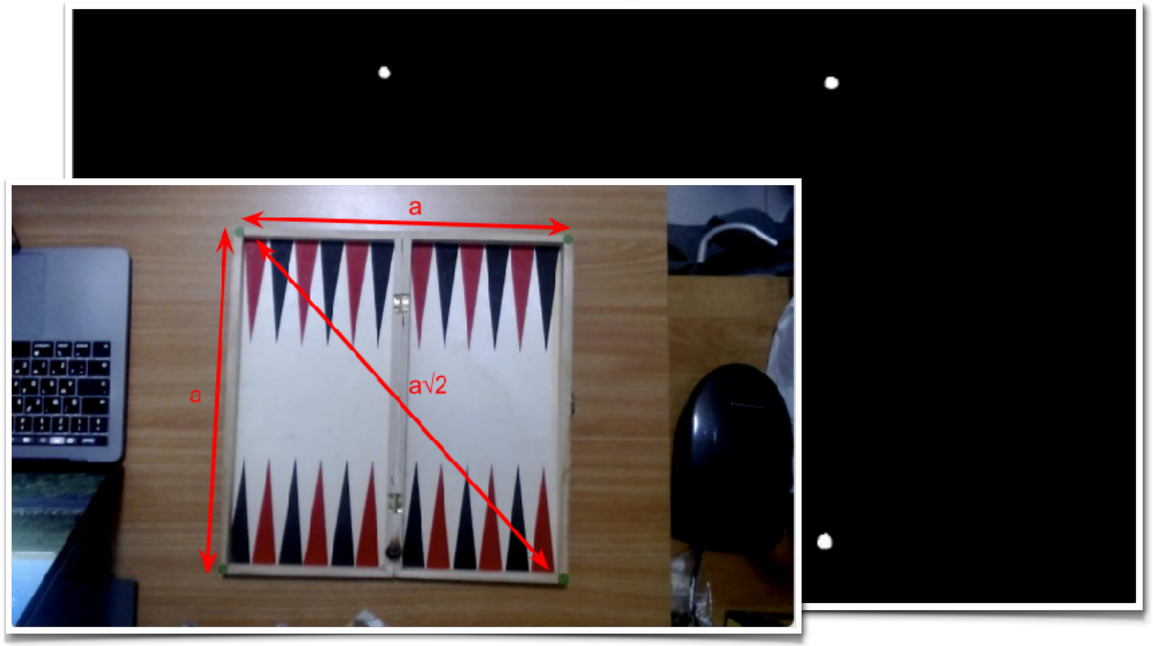
# Board Detection, Take #1

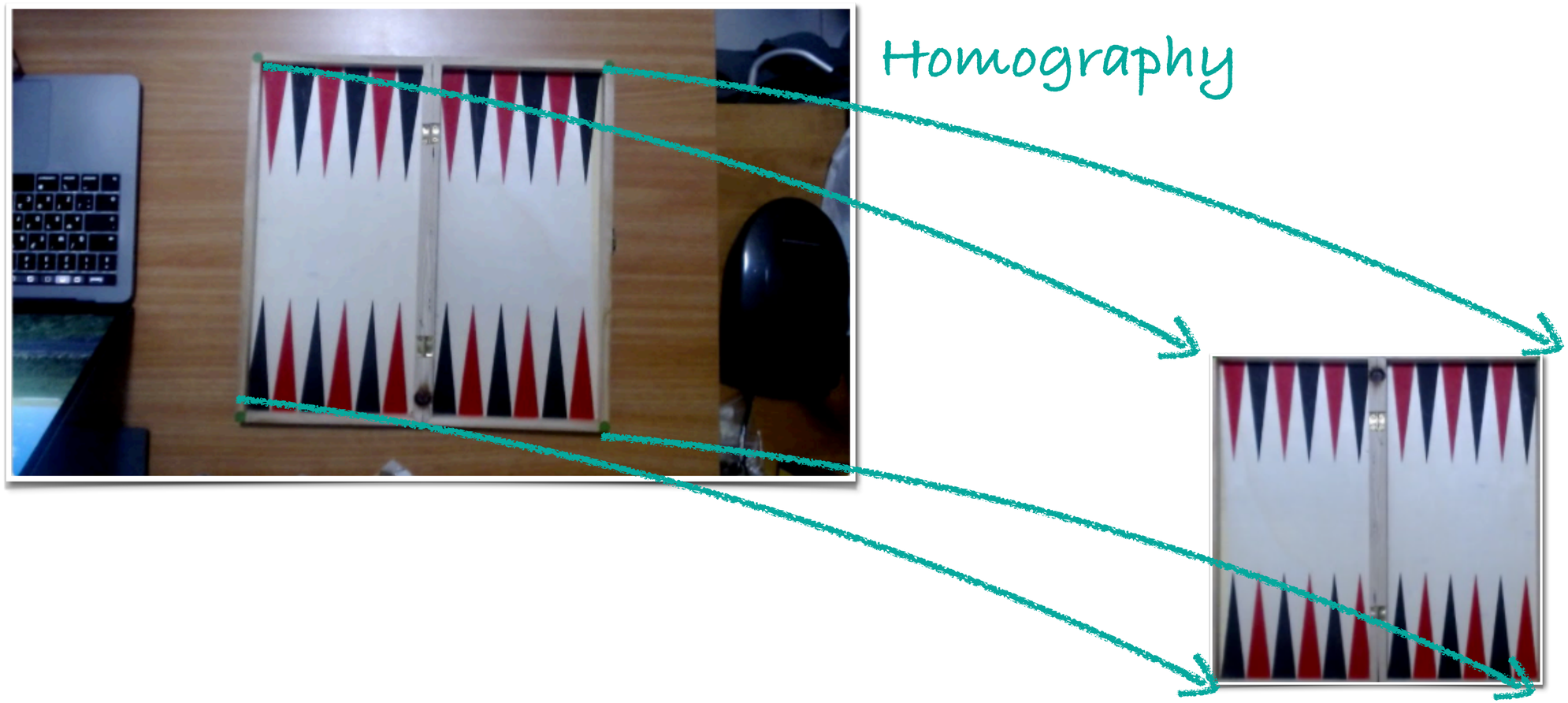
- **Requirements**
  - Needs to be highly accurate
  - Needs to include alignment of the board
  - Must not rely on visual features of the board (covered during the game)
- **Challenges**
  - Colors issue — cannot effectively distinct the board from it’s surrounding
  - Spent days on this, in vain 😟



# Board Detection, Take #2

- Added 4 circle-shaped green staples on the board's vertices
- **Detection**:
  - Green HSV filter
  - Erosion and dilation
  - Hough circle transform
  - (Optional) geometric verification, with some slack for camera angle flexibility






# Board Alignment using Homography

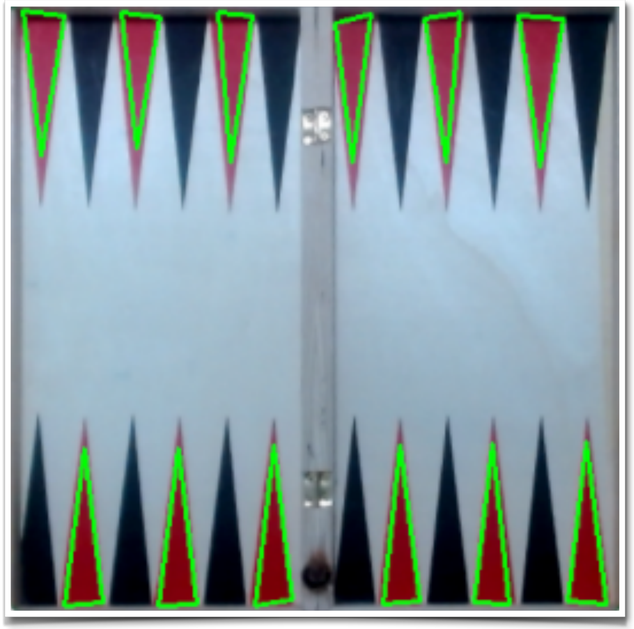
Original image to 400x400 tightly cropped and aligned board

# Board Calibration: Goals

- Find all areas on the board where checkers can “legally” be placed
- Define them numerically on the tightly cropped board image: ***Checkers Containers***
- Automatic process with no user intervention
- Done once, before the game starts; on an **empty board**
- During the game, detect the location of checkers and associate them to containers

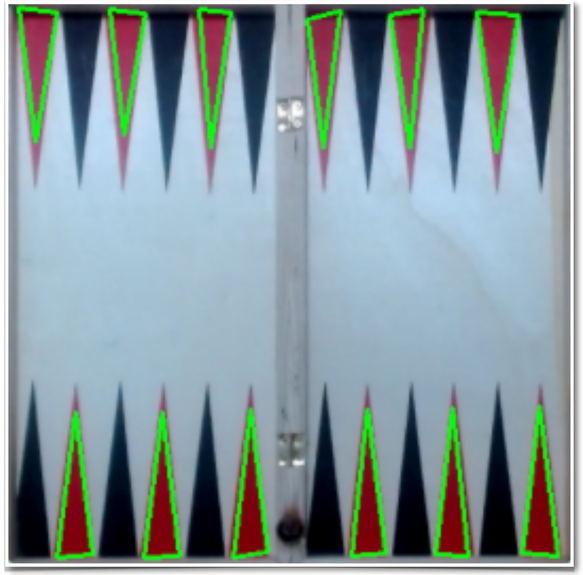
# Calibration Step #1

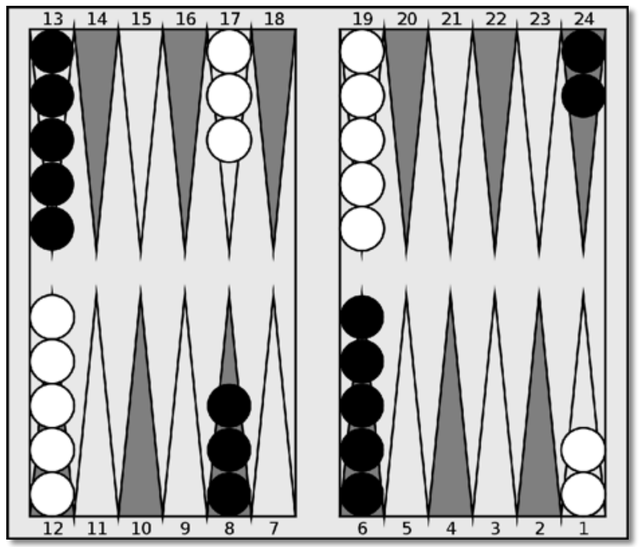
- The red triangles are most distinct in color and shape
- **Triangle Detection:**
  - Gaussian blur
  - Red HSV filter
  - Filter all contours with 3 edges
  - Filter triangles with ratio 1:4.5 base to edges

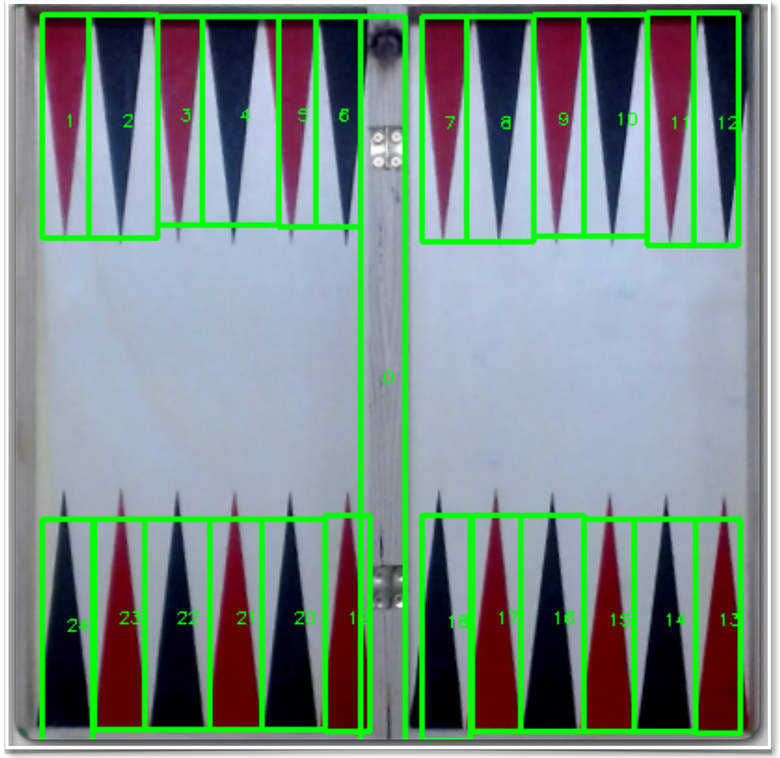


# Calibration Step #2

- Create all Checkers Containers
  - Create bounding box around each red triangle (12)
  - Sort bounding boxes top-to-bottom, left-to-right (6 top, 6 bottom)
  - Calculate bounding box for all black rectangles (12)
  - Calculate bounding box for the bar (1)
  - The order is reverse as the camera opposes the player

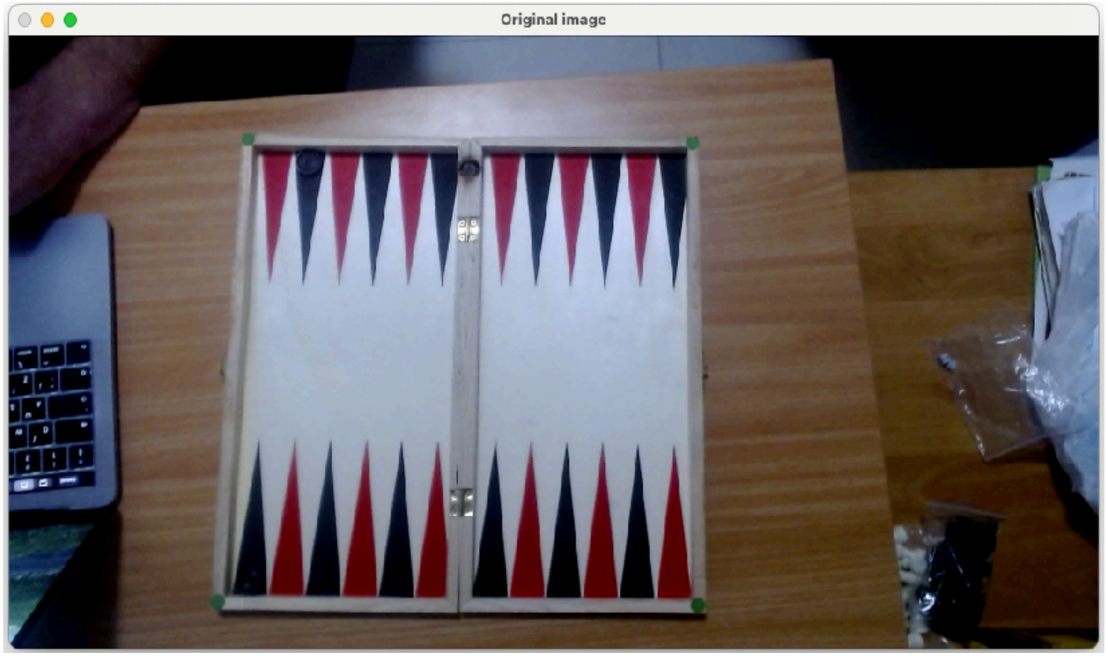


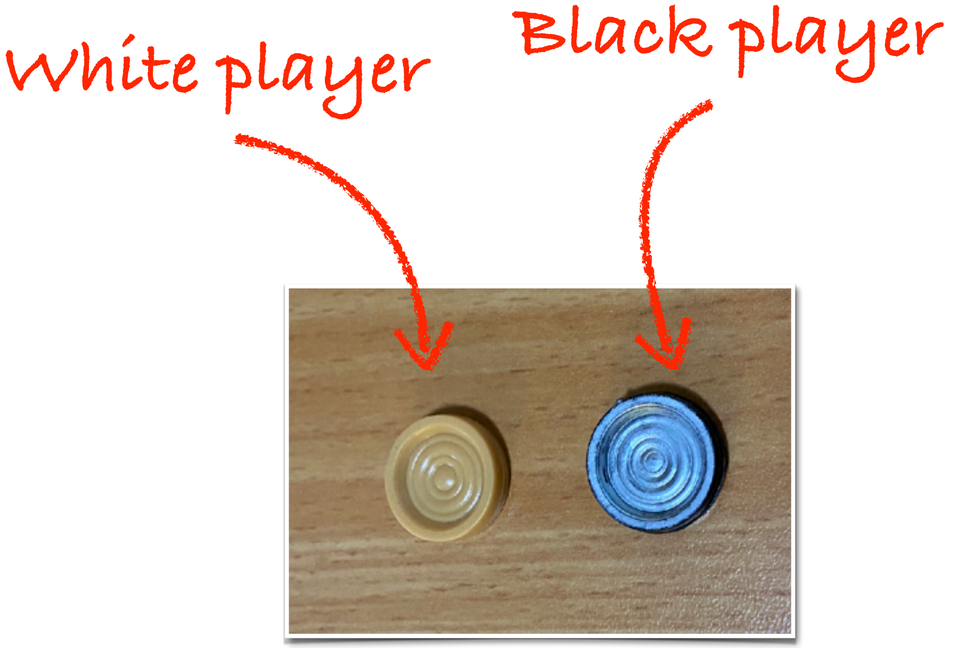


# During the Game: Detect Current Board Status

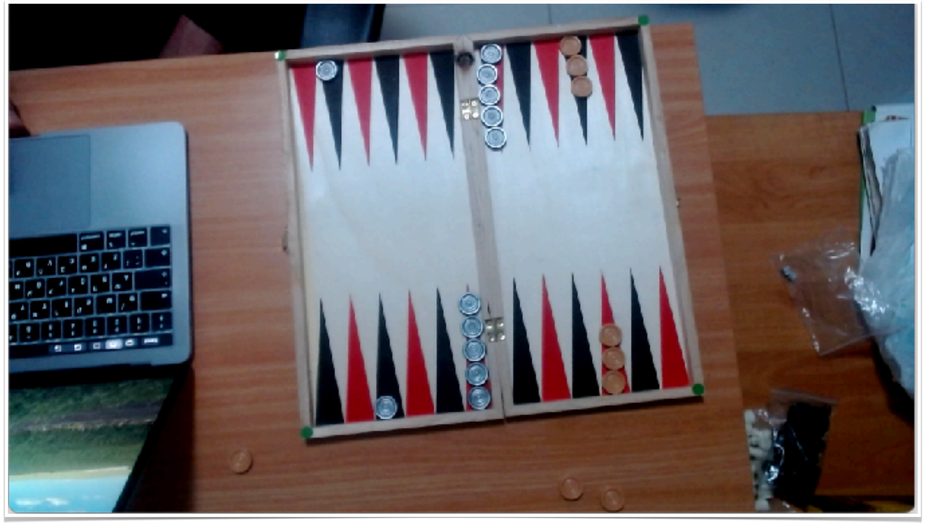
- Again, color issues!
- Can you spot the 2 black checkers on the board?

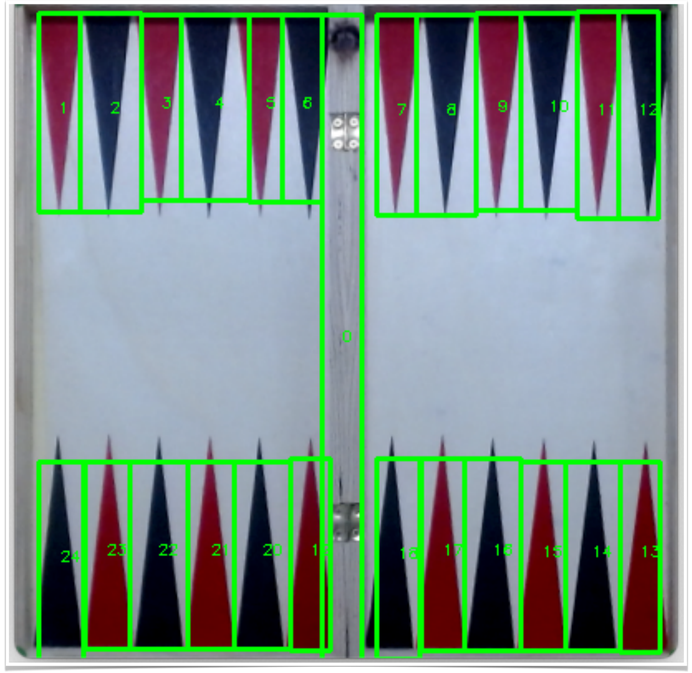


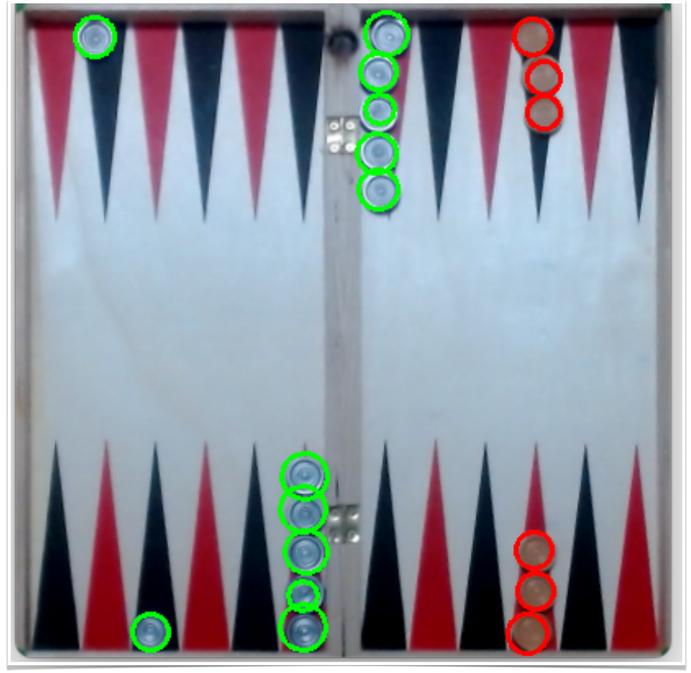



# Detect Current Board Status

- Apply 2 HSV filters (caramel & silver)
- Erosion and dilation
- Hough circles transform
- Associate each circle’s center to it’s respective checker container box







# Summary

- It was a great experience in realizing the challenges of computer vision
- There is a real challenge in developing **robust** computer vision software
- We liked the outcome
  - Backgammon is a fun game to play
  - Good user experience with adequate performance
  - Need good lighting conditions!