Description:

We seek to use the techniques outlined in class to develop a basic augmented reality demo on an Android smartphone. Ultimately we will demo an augmented reality Tic Tac Toe game.

We envision a user drawing a Tic Tac Toe board on a sheet of paper. The user will mark his or her moves with pen or pencil. The computer will mark its moves digitally and project them on top of the smartphone camera's image of the board. In a sense, the Android smartphone will act as a lens into the game world. We will initially develop an app that works on head on, unrotated game boards, and we hope to go on to enable the app to view the board from any orientation.

We choose this project because it uses a wide scope of image processing techniques. We will first have to prepare the image for detection, run feature detection, and then extract important information from our features regarding the state of the board. Beyond detecting the board and moves themselves we will have to detect orientation to establish the 2D plane the board lies on in 3D space and the camera's position relative to this plane. This information will aid in our later task of performing a transformation to map our virtual images to the game board. Finally, after the projection, we must run fast image alignment using our feature detector to really reinforce the notion of augmented reality.

Milestones:

Orthogonal orientation

1. Use feature detection to extract the Tic Tac Toe board as well as the human player's moves
2. From the extracted Tic Tac Toe board and player moves establish the current game state
3. Write an alpha beta game player to compute the computer's move based on the current game state
4. Project the computer's move onto the camera's current view of the real game

Affine Transformations
5. Repeat milestone 1 to account for an affine transformation. This will allow us to extract the board and player moves independent of phone orientation.
6. Repeat milestone 4 to account for arbitrary phone orientations.

Additional (Independent of Augmented Reality):

7. Replace the alpha beta player with a tiered difficulty player
   a. Easy: random
   b. Medium: depth 2 alpha beta
   c. Expert: Alpha Beta with caching

* These milestones ensure that we will at least have a working demo

**References:**

[1] “Feature Detection and Description in OpenCV.”
http://docs.opencv.org/modules/nonfree/doc/feature_detection.html

http://docs.opencv.org/doc/tutorials/core/basic_geometric_drawing/basic_geometric_drawing.html

[3] Simple Augmented Reality SDK for OPENCV

http://www.stanford.edu/class/ee368/Project_13/Reports/Naqvi_Sikora_AR_Equation_Plotter.pdf

http://www.stanford.edu/class/ee368/Project_12/Reports/Tian_Meng_A_Virtual_Telescope_for_Augmented_Reality_Application.pdf

**Requires Android**