**Tic Tac Toe with OpenCV**

**and Dobot Magician**

CSCI-C458 Intelligent Robots

Semester Project

Zac Clifton:

Cody Martin:

martincody423@gmail.com

Motivation: The purpose of this project ties together the study and manipulation of a robotic arm that is not well documented through the use of sensor and effector. We study and implement search algorithm to create an autonomous robotic player for the game Tic-Tac-Toe (TTT).

Project Goals: At the onset, we create the game board. We use the symbols ‘O’ and ‘X’ as superficial tokens on the game board. Then we look at the manipulation of the hardware of our project. We create the ‘X’ token with robotic arm and ink pen effector. Then we create an ‘X’ on every cell in the game board. [Camera sensor work]

Now we create the TTT game logic from which our robot corresponds as a player. We find a suitable search algorithm to achieve autonomy of A.I. player.

Intended Approach: We set up the game environment (board) with an internal program of the physical locations of the vertices of the game board, with respect to the robotic arm. We then define general offset values from each vertex *inside* each cell as the four coordinates necessary to create an ‘X’ in each game cell.

We create the definition of a TTT game with the appropriate start state and end states. To progress the state of the game, we have game logic implemented through the use of a search algorithm. The algorithm compares a heuristic value that represents the outcome of the game as the program simulates each possible move from the current state. To update the internal state of the game board for the non-robotic (human) player, we use a camera sensor. [Camera sensor approach]

Test Bed: We use Dobot Magician with an ink pen effector to draw the ‘X’ token on the game board as our robotic player. [Camera info] We communicate between camera, internal game logic, and Dobot entirely through Python language.

Evaluation: For testing and implementation, we consider the size of the ‘X’ token to be within the area of the corresponding cell, to allow for error as we manually calibrate Dobot, camera, and game board. To test the game logic, we run several cases in which human or Dobot wins on each subsequent turn after turn 4 (as a winning state can only happen on turn 5 or above). We test that A.I. robotic player recognizes a draw state, win state, and loss state and subsequently makes no further moves. Specifically, we use unit tests to evaluate the correctness of the search algorithm and discover and patch errors in logic. [Further testing]

Milestones:

Find vertices of game board: Week of 10 March

Create ‘X’ token: Week of 10 March

Create TTT Game Environment: Week of 24 March

Implement Search Algorithm: Week of 31 March

Project Presentation: 19 April