Machine Vision and Alternative Game Control

New Mexico Supercomputing Challenge
Final Report
April 1, 2009

Team #106
Monte del Sol Charter School

Team Members
Tenzin Lekden Lungtok
Cole Tuffli

Sponsoring Teacher
Natalie Martino

Mentors
Stephen Guerin
John Paul Gonzales

Executive Summary

This is a project designed to demonstrate alternatives to game control and human interface devices using machine vision. Machine vision allows computers to interpret things like color, shape and motion and translate them into functions that are normally performed with a mouse and keyboard. Using machine vision we are able to explore new ways of interacting with computer programs and effectively take the program out of the computer.

In order to explore machine vision we decided on a simple interactive game interface. Pong (Table Tennis or Air Hockey) was chosen because it was a simple, well understood program and presented the easiest game to recreate. Machine vision algorithms detecting infrared surface points were used to interface control between computer and a projected playing field. This "ambient computing" presents users a new way of interacting with computers and may serve serve to redefine the relationship lay users have with computers and, until now, flat screen boxed displays.

General Outline

We started out with some basic idea of a game we can design on Netlogo, which is a multi-agent programming language and integrated modeling environment. We came up with several ideas, such as a dart game, painting, fireworks, and air hockey. We thought a air hockey game would be a interesting and simple to make on Netlogo. It would challenge us as students and mentors on how we put the code down to make this environment.

We went through some problems on how we would lay down this game environment in Netlogo, considering that NetLogo is designed to be a patches-and-agent behavior modeling environment. We also decided to incorporate
player control into the NetLogo program in order to make the game fun to play.
We decided to make the puck a turtle since the puck would be moving around trying to score in the opponent goal, which is the main point of the game. Then we used the patches to lay our environment down such as the walls, the goal area, the staring point of the puck and player sides. When ever the puck passes over a set colored patch, it activates a goal scored.

We had problem making it two player game because we couldn't control the other paddle without distracting the player 1 paddle. We thought it would be better to add a computer player. The computer player is programed to follow the movement of the puck along the Y-axis in the world with a fixed speed in order to give the game the feel of the original pong. We also added the difficulty of the computer by setting how fast and slow the opponent paddle would move.

After we got the game running, we used a projector to project it down to a surface. Then we use custom-built infrared LED "pucks" to control our program. When ever the real infrared paddle moves up and down, the virtual paddle in our program also follows along with it.

Background to Pong/Air Hockey

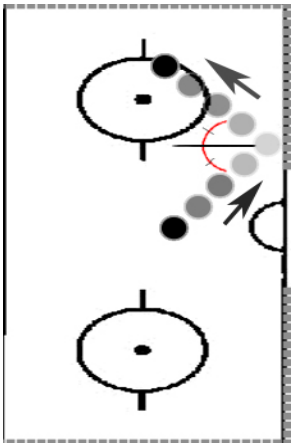
Pong was one of the first computer games made. It was based on the idea of an electronic ping-pong game and was created in 1972 by Allan Alcorn. The idea of the game is to get the highest score by making your opponent miss the ball. You could play against a computer or a second player person. The game pong was the founding game that started the gaming industry. In 1975 the game pong was in stores for an at-home version for purchase for the Christmas session. Sears was the only store that carried pong at this time. The game was a commercial success and led to numerous copies

Air hockey, like pong is a game for two competing players trying to score points in the opposing player's goal. In 1972 a billiards employee named Bob Lemieux claims that he invited the game of air hockey. Air Hockey was actually invented by a trio of Brunswick engineers: Phil Crossman, Bob Kendrick, and Brad Baldwin. The trio was attempting to create a game utilizing a frictionless surface. The trio created the game in 1969 3 years before Bob Lemieux claimed he had made it.

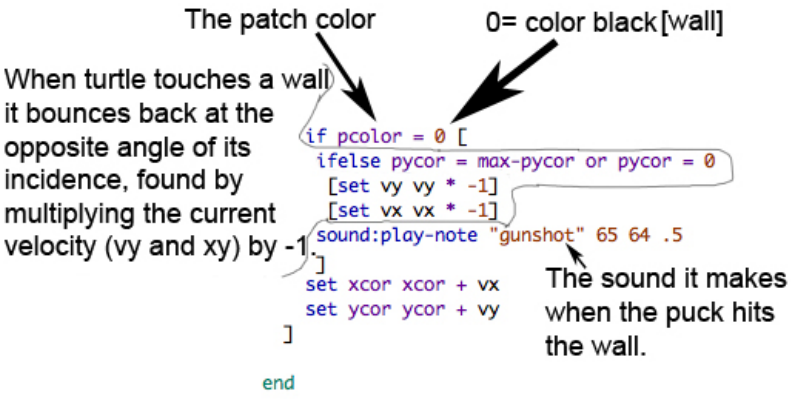
Mathematical and Physical Formulas

The workhorse behind the Pong program is a simple law of physics that allows us to program bounce behavior: the angle of reflection θ_r on a smooth surface is equal to the angle of incidence θ_a:

$$\theta_r = \theta_a$$



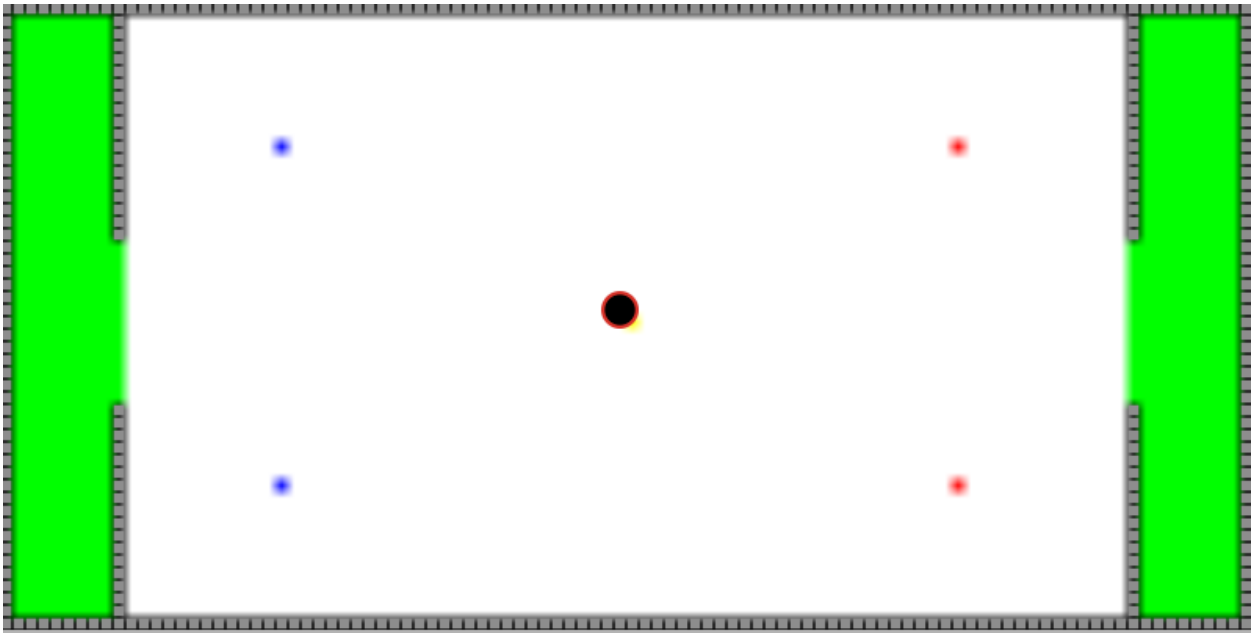
In our code, this is done by taking our initial velocities in the x- and y-directions (vx and vy) and applying a simple mathematical operation to them: multiplication by -1. This effectively reflects the puck back in the opposite direction of incidence, allowing for proper puck behavior in the ideal environment of the computer program.



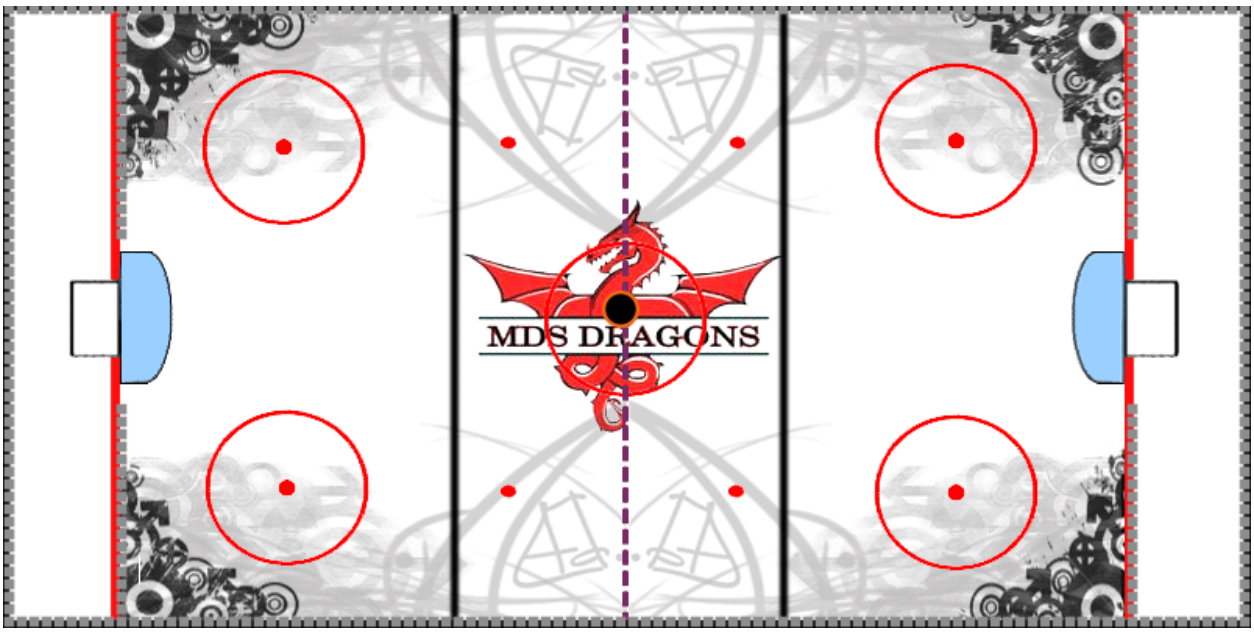
Approach

We started off with the basic ice a hockey rink. From there we added colors to help us determine were are main points are going to be.

- Goals (green)
- Walls (black)
- Staring point of the puck (yellow, obscured by puck)
- Player sides (player 1[red] player 2[blue])



Once we created the background, we created walls using turtles. The turtles were instructed to sprout on patches with pcolor=0 (gray) and set their own color to 0. The walls were then programmed to interact with the puck when the puck "steps" on their patch.



 Once the walls were in place, we went through photoshop and made a background that uses the same forms of a hockey rink. We also included our school mascot into the middle to show our school spirit.

Once we created the walls and the background we stated on the puck. We made it so the puck would always start in the middle of the rink. Once the puck was made we gave it the command that when the player presses go it would travel in a random direction across the x- and y- axes. We also programed it to have a random velocity. It is unpredictable witch way and how fast the puck will travel from the origin, making the game fast-paced and frantic or slow and strategic.

Once the puck was operational and moving we created the goal and scoreboard. We used the same background we used for the walls to make the goals. We programed the goals to give a point to a player once the puck passes the designated goal area of the Y-axis. Once the puck scores a goal, it resets to the center with a new velocity and heading. We programed the scoreboards to add one point to a player depending on which goal is passed (player 2 scores when the puck enters the left goal, player 1 scores when the puck enters the right goal).

Once we had the scoreboard we added basic sound for goals and when the puck bounced of a wall.

After we added the scoreboard we went in to make the paddles. We started of with the player one's paddle. (red) we set the paddle to middle of the red side. we set up the paddle to move with the mouse witch in Netlogo is called mouse down. this aloud us to move the paddle by moving our mouse up or down, left and right. we made the paddle have hit detection. this made it so

that when the puck would hit the paddle it would use the pucks velocity and bounce it of in the opposite direction. and also increased the velocity to make the game play harder to keep going for long periods of time.

Once we had the human player added in and finished we added the A.I person. the blue player was created to give players someone to play. we stated by putting in a new paddle but on the other side of red (the blue side). we set the turtle to fallow the puck were ever it would go. the way we did this is we used the pucks velocity to determine were it was going to go on the x-y axis and had the paddle meat it there. the only problem was that the puck couldn't move as fast as the puck so the puck would hit the wall and bounce of witch would change were it was going to end up on the x-y axis. after we hit that part we made possible ranges on how fast the A.I could change course with the puck.

now our game is ready to be played.

Wii Remote

The Wii Remote is the most important part of the machine vision component of our project. The Wii Remote (or "Wiimote") is basically an infrared sensing camera that is able to pinpoint infrared source locations. For the machine vision systems to work, it was the easiest and most effective tool we could find. In addition, we used a program called Wiimote Whiteboard that allowed us to connect a wiimote to a computer using the Bluetooth and use the Wiimotevirual pucks on the field. The game can then be played normally.
sensors to detect infrared sources. When a projector is used to project a playing field onto a surface, we set up our real-world pucks in alignment with the

Results

We've managed to develop a fully playable game that can be played on a computer, a TV screen or on a table using a projector. We were able to set up the Wii remote so that it could sense an infrared light source in order to track . we also were able to set up our web cam to use machine vision to sense the movements of the players hands. We were able to make the A.I player with multiple difficulty levels. All four of us feel that this is a very strong beginners game that adds in multiple different ways to play games.

The game itself would not be difficult to expand to any kind of surface. A virtual pong that senses the position of salt-and-pepper shakers on a table while patrons are waiting for a meal is easily within sight. The idea of interaction with computers using machine vision on any surface is an idea that is just beginning to be explored.

Program codes ((not done))

```
extensions [sound]

globals [player1Score player2score]

breed [pucks puck]
breed [walls wall]
breed ]paddles paddle]

pucks-own [vx vy]
```

ornage is our globals. that mean that these are the things the need to be defined because Netlogo does not know these words. breeds are our types of turtles that get defined in the code. walls, pucks and paddles are our turtles that we need Netlogo to know. vx vy are our velocity going in the x and y axis.

```
to setup
  clear-all
  import-drawing "arena.png"
  import-pcolors "hockeyPatches.png"
  create-pucks 1 [
    set shape "puck"
    set size 3
    setxy 49 25
    set vx random-float velocity - (.5 * velocity)
    set vy random-float velocity - (.5 * velocity)
  set player1Score 0
  set player2Score 0
  ask patches with [pcolor = 0][sprout-walls 1 [set shape "square"  set
heading 0  set color gray]]
  create-paddles 1 [set shape "square" setxy 85 25  set size 2 set color blue
set bluepaddle self]
  create-paddles 1 [set shape "square" setxy 15 25  set size 2 set color red
set redPaddle self]


end
```

Green is our to setup stage that's were when someone clicks the button on the program that sets up the game so it can be played. clear-all is so that if there is a game going on it clears everything like the scoreboard. and any movement of the puck. import-drawing is were we added our graphics to

give us our arena patches and our cool looking background. create-pucks 1 is our puck we set the shape to puck with is a shape we created in the Netlogo program. we created the size to make it look better with the size of our game. random-float velocity - (.5 * velocity) is so that once it starts it is in the middle but is going a random way every time. sprout-walls we told the patches that were black to sprout walls and we gave them the shape of 3D squares and we had them set there color to gray. create-paddles 1 is us creating our paddles giving them shape size and color. it also gives them the the command on were to set them selfs on the board.

```
to go
  move-pucks
 move-paddle
 move-pad
  tick
end
```

Blue is our to go stage this is what happens when the player presses go on the game to get ready to play. move-pucks, move-paddle, move-pad (defined below) is set on ticks with is how Netlogo moves things in the program so the puck moves with the ticks. make ticks faster puck goes faster same with when you turn the ticks down.

```
to move-pucks
  ask pucks [
    if pcolor = 64.9 [
      ifelse xcor < 50
       [set player2score player2Score + 1]
       [set player1score player1Score + 1]
     setxy 49 25
       set vx random-float velocity - (.5 * velocity)
       set vy random-float velocity - (.5 * velocity)
      sound:play-note "Music Box" 65 64 .5
       sound:play-note-later 1 "Applause" 70 64 1

    ]


    if pcolor = 0
[
     ifelse pycor = max-pycor or pycor = 0
      [set vy vy * -1]
      [set vx vx * -1]
     sound:play-note "gunshot" 65 64 .5
```

```
  ]
   set xcor xcor + vx
   set ycor ycor + vy
 ]

end
```

Red is our to move-pucks stage witch tells the pucks how to move through the game. ask pucks [
    if pcolor = 64.9 [
      ifelse xcor < 50
       [set player2score player2Score + 1]
       [set player1score player1Score + 1] this is the score part of the game telling to puck that once it hit past this point o make the sound and to give one player the point are the scoreboard.
    setxy 49 25
       set vx random-float 1
      set vy random-float 1
      sound:play-note "Music Box" 65 64 .5
       sound:play-note-later 1 "Applause" 70 64 1

    ] this is the part with the puck resetting itself after someone gets a goal.
set xy 49 25 this is telling the puck to reset itself at x-49 and y-25.
set vx random-float velocity - (.5 * velocity)
      set vy random-float velocity - (.5 * velocity) this is telling it to start in a random direction in the x and y with a .5 times on its velocity.
sound:play-note "Music Box" 65 64 .5
       sound:play-note-later 1 "Applause" 70 64 1
 this is the sound that it makes when someone gets a goal. it plays .5 and 1 second seconds.

if pcolor = 0 pcolor is short for patch color and 0 is black.
[
     ifelse pycor = max-pycor or pycor = 0
      [set vy vy * -1]
      [set vx vx * -1]
     sound:play-note "gunshot" 65 64 .5
this is stating that if a turtle hits the pycor of black to bounce if with the direction of the puck but staying in the same way.
   ]
    set xcor xcor + vx
    set ycor ycor + vy

  ] this is telling the puck to set xcor and ycor plus what the velocity.
end
```
```

```
to move-paddles
  ask redPaddle [
    set ycor mouse-ycor
    set xcor mouse-xcor
    ask pucks [if distance myself < [size] of myself [set vx vx * -1]]
  ]
end
```

to move-paddles this is telling the paddles how to move.

ask redPaddle [we are only talking to the red paddle right now

set ycor mouse-ycor
set xcor mouse-xcor this is telling the red paddle to set its self were my (the players) mouse is. this alows the player to control the puck with the mouse.
    ask pucks [if distance myself < [size] of myself [set vx vx * -1]] this is the part were it is telling the puck that if it hits the paddle to bounce off in the other direction loosing parts of its velocity.
  ]
end

```
to move-pad
  ask bluepaddle [
    if (ycor - [ycor] of one-of pucks < -1) [set ycor ycor + expert-level / 10]
    if (ycor - [ycor] of one-of pucks > 1) [set ycor ycor - expert-level / 10]
    ask pucks [if distance myself < [size] of myself [set vx vx * -1]]
```

to move-pad new term just for the blue paddle.
  ask bluepaddle [

 if (ycor - [ycor] of one-of pucks < -1) [set ycor ycor + expert-level / 10]
    if (ycor - [ycor] of one-of pucks > 1) [set ycor ycor - expert-level / 10]
this whole part is telling the blue paddle to go to were the puck is going on the y axis. this alows us to create levels by changing the spped of the puck so that once the puck hits the wall the paddle would be able to fallow faster.

    ask pucks [if distance myself < [size] of myself [set vx vx * -1]] this is the part were it is telling the puck that if it hits the paddle to bounce off in the other direction loosing parts of its velocity.

[[[[[[]]]]]]] in netlogo everything has to be in [[[]]]] so the computer can know what is being defined.

END is a part of the code that has to be added it is telling the program that that is the end of the command so that you can start a new one.

Conclusion

our project went well and we feel that over all it came out very well. with this project we think we could expand or air hockey game further by doing multiple things that could be fun. one thing we had in mind is go to a ice rank and project our game on to it so we could play it as the players but be on ice skates. we also have the idea of using the machine vision to find a person a a floor and play the game with our feet. we also thought about getting rid of the A.I and be able to play someone else to increase the playing type.

Bibliography

"Air Hockey." Wikipedia, The Free Encyclopedia. 25 March 2009, at 14:02. Wikimedia Foundation, Inc. 26 March 2009 <http://en.wikipedia.org/wiki/air_hockey>

Young, Hung. University Physics: Eight Editions. Reading Massachuestts: Addison-Wesley Publishing Company, 1992

Lee, Johnny Chung. "Low-cost Multi-point Interactive Whiteboards Using the Wiimote." Johnny Chung Lee Projects. 2008. Accessed 10 February 2009. <http://johnnylee.net/projects/wii/>

"Pong." Wikipedia, The Free Encyclopedia. 25 March 2009, at 13:57. Wikimedia Foundation, Inc. 26 March 2009 <http://en.wikipedia.org/wiki/Pong>

Young, Hung. University Physics: Eight Editions. Reading Massachuestts: Addison-Wesley Publishing Company, 1992

Wilensky, Uri. "Netlogo." Netlogo. 1999–2009. Northwestern University 2 January 2009. <http://ccl.northwestern.edu/netlogo/>.

Appendix A: Creation of Graphics

We used Adobe Photoshop to make all the graphics in the game. We have two different layers set on Netlogo. One would be the patch colors, which helps on the action when the puck touches the colors. The second layer which is over the patch layer, is just the design of the arena. This layer is just for looks. The patch layer is the most important because it is layer that talks back and forward to Netlogo and the codes telling each other what to do.

Thanks to

We would like to thank our mentors (team leaders) Stephen Guerin, and John Paul Gonzales. Without them we would have never joined the challenge or even learn how to create a game using Netlogo. We would also like to thank our team sponsor Natalie Martino. Also, thanks to every person who has put money into making this challenge possible for students from across the state to compete in. Also, we can't forget our school. If it wasn't for our school and our mentorship program we would have never met our mentors or even been told about the challenge.