/\* Daniel MacMillan and Dangnhi Ngo

\*

\* Lab 5

\* Servo motor and sensor to detect open space

\*/

#include <kipr/botball.h>

#define VEL 50

void move\_servo(int servo, int pos)

{

int newpos;

enable\_servos();

set\_servo\_position(servo, pos);

msleep(600);

disable\_servos();

msleep(600);

newpos = get\_servo\_position(servo);

printf("%d\n", newpos);

}

int compare\_distance(int farthest\_reading, int next\_pos, int\* farthest)

{

int current\_reading = analog(0);

if(current\_reading < farthest\_reading)

{

printf("new farthest: %d -> %d\n", \*farthest, next\_pos);

\*farthest = next\_pos;

return current\_reading;

}

return farthest\_reading;

}

//function to make the robot go straight

void go\_straight()

{

motor(0,VEL/2\*3);

motor(1,VEL/2\*3);

printf("go straight\n");

msleep(1000);

}

void move\_robot(int farthest)

{

printf("choosing this angle to move towards: %d\n", farthest);

if(farthest == 0)

{

// turn left

motor(0, VEL);

motor(1, -VEL);

msleep(2000);

go\_straight();

}

else if(farthest == 512)

{

// turn left-ish

motor(0, VEL);

motor(1, -VEL/2);

msleep(2000);

go\_straight();

}

else if(farthest == 1024)

{

// go forward

go\_straight();

}

else if(farthest == 1536)

{

// turn right-ish

motor(0, -VEL/2);

motor(1, VEL);

msleep(2000);

go\_straight();

}

else if(farthest == 2047)

{

// turn right

motor(0, -VEL);

motor(1, VEL);

msleep(2000);

go\_straight();

}

else

{

printf("This isn't a valid direction: %d\n", farthest);

}

motor(0, 0);

motor(1, 0);

return;

}

int main()

{

int farthest\_reading;

int farthest;

set\_servo\_position(0,0);

while(!c\_button\_clicked())

{

// reset readings

farthest\_reading = 6000;

farthest = 0;

// move the sensor around and sense in 5 different places

move\_servo(0, 0);

farthest\_reading = compare\_distance(farthest\_reading, 0, &farthest);

move\_servo(0, 512);

farthest\_reading = compare\_distance(farthest\_reading, 512, &farthest);

move\_servo(0, 1024);

farthest\_reading = compare\_distance(farthest\_reading, 1024, &farthest);

move\_servo(0, 1536);

farthest\_reading = compare\_distance(farthest\_reading, 1536, &farthest);

move\_servo(0, 2047);

farthest\_reading = compare\_distance(farthest\_reading, 2047, &farthest);

// move in the direction of farthest

move\_robot(farthest);

printf("reset...\n");

}

disable\_servos();

ao();

return 0;

}