%function [] = economy\_TEAM(floor\_plan)

floorplan = importdata("floor\_plan\_3.floor");

%=======================================================================

%INITIALIZATION OF VARIABLES

%initial robot coordinates set

roombaX = 2;

roombaY = 2;

%initial move direction is set (0 = does not move)

movedir = 0;

%coordinates of the charging pad

chargeX = 2;

chargeY = 2;

%create copy of floorplan to track where the robot has been (marked as 5)

clean\_map = floorplan;

%variable set to 1 when direction needs to be changed

switch\_dir = 0;

%variable set to 1 when charge is needed

need\_charge = 0;

%i represents the battery life, the robot moves one tile per loop

% change below to just check for 3's with "high-end" model

while ismember(3, clean\_map) == 1 || ismember(4, clean\_map) == 1

%=======================================================================

%FIND NEXT UNCLEAN LOCATION - starts path from current position

for i = 1:250 %change to 350 for "regular" model

%find next tile to be cleaned if not returning to charger

if need\_charge == 0

%change below to [k,m] = find(clean\_map == 3,1); for "high-end"

%model

[k,m] = find(clean\_map < 5 & clean\_map > 2,1);

end

%=======================================================================

%MOVEMENT - moves the robot by setting previous tile to white and

%setting current tile red.

switch movedir

case 1 % up

roombaY = roombaY - 1;

floorplan(roombaY + 1, roombaX) = 4;

clean\_map(roombaY + 1, roombaX) = 5;

floorplan(roombaY, roombaX) = 2;

case 2 % right

roombaX = roombaX + 1;

floorplan(roombaY, roombaX - 1) = 4;

clean\_map(roombaY, roombaX - 1) = 5;

floorplan(roombaY, roombaX) = 2;

case 3 % down

roombaY = roombaY + 1;

floorplan(roombaY - 1, roombaX) = 4;

clean\_map(roombaY - 1, roombaX) = 5;

floorplan(roombaY, roombaX) = 2;

case 4 % left

roombaX = roombaX - 1;

floorplan(roombaY, roombaX + 1) = 4;

clean\_map(roombaY, roombaX + 1) = 5;

floorplan(roombaY, roombaX) = 2;

otherwise

end

%=========================================================================

%MOVEMENT DIRECTIONS - declares what movement the robot will have

%next loop

%time between each movement in seconds

pause(.005);

%makes tile charging pad tile become blue whenever robot is off of it

if chargeX ~= roombaX || chargeY ~= roombaY

floorplan(chargeY, chargeX) = 5;

end

%displays the new floorplan before next move is determined

dispfloor(floorplan);

%function used to find next direction of movement

[movedir, switch\_dir, clean\_map] = nextdir(k, m, roombaX, roombaY, movedir, switch\_dir, floorplan, clean\_map, need\_charge);

end

%=======================================================================

%RETURN TO CHARGE

%returns to charging pad

movedir = 0;

k = chargeY;

m = chargeX;

need\_charge = 1;

%find route and travel back to charge station

while need\_charge == 1

[movedir, switch\_dir, clean\_map, need\_charge] = nextdir(k, m, roombaX, roombaY, movedir, switch\_dir, floorplan, clean\_map, need\_charge);

switch movedir

case 1 % up

roombaY = roombaY - 1;

floorplan(roombaY + 1, roombaX) = 4;

floorplan(roombaY, roombaX) = 2;

case 2 % right

roombaX = roombaX + 1;

floorplan(roombaY, roombaX - 1) = 4;

floorplan(roombaY, roombaX) = 2;

case 3 % down

roombaY = roombaY + 1;

floorplan(roombaY - 1, roombaX) = 4;

floorplan(roombaY, roombaX) = 2;

case 4 % left

roombaX = roombaX - 1;

floorplan(roombaY, roombaX + 1) = 4;

floorplan(roombaY, roombaX) = 2;

otherwise

end

%time between each movement in seconds

pause(.005);

%displays the new floorplan

dispfloor(floorplan);

end

end

function [movedir, switch\_dir, clean\_map, need\_charge] = nextdir(k, m, roombaX, roombaY, movedir, switch\_dir, floorplan, clean\_map, need\_charge)

%if in desired position to clean mark as cleaned or reset charge

if roombaX == m && roombaY == k

movedir = 0;

if need\_charge == 1;

need\_charge = 0;

else

clean\_map(roombaY, roombaX) = 5;

end

%-----------------------------------------------------RIGHT MOVEMENT----------------------------------------------------

%move right if no obstacle

elseif roombaX < m && floorplan(roombaY, roombaX + 1) > 2

movedir = 2;

if switch\_dir == 1

switch\_dir = 0;

end

%move up if no obstacle is up and right is obstacle

elseif roombaX < m && switch\_dir == 0

if floorplan(roombaY - 1, roombaX) > 2

movedir = 1;

else

switch\_dir = 1;

movedir = 0;

end

%move down if obstacle up and right (switch directions)

elseif roombaX < m && floorplan(roombaY + 1, roombaX) > 2 && switch\_dir == 1

movedir = 3;

%move left if up, down, and right have obstacles

elseif roombaX < m && roombaY >= k && floorplan(roombaY - 1, roombaX) < 3 && floorplan(roombaY, roombaX - 1) > 2

movedir = 4;

%------------------------------------------------------LEFT MOVEMENT-----------------------------------------------------

%move left if no obstacle

elseif roombaX > m && floorplan(roombaY, roombaX - 1) > 2 && movedir ~= 2

movedir = 4;

if switch\_dir == 2

switch\_dir = 0;

end

%move up if no obstacle is up and left is obstacle

elseif roombaX > m && switch\_dir == 0

if floorplan(roombaY - 1, roombaX) > 2

movedir = 1;

else

switch\_dir = 2;

movedir = 0;

end

%move down if obstacle up and left (switch directions)

elseif roombaX > m && floorplan(roombaY + 1, roombaX) > 2 && switch\_dir == 2

movedir = 3;

%move right if up, down, and left have obstacles

elseif roombaX > m && roombaY > k && floorplan(roombaY - 1, roombaX) < 3 && floorplan(roombaY, roombaX + 1) > 2

movedir = 2;

%--------------------------------------------------------UP MOVEMENT-------------------------------------------------------

%move up if no obstacle

elseif roombaY > k && floorplan(roombaY - 1, roombaX) > 2

movedir = 1;

if switch\_dir == 3

switch\_dir = 0;

end

%move right if no obstacle is right and up is obstacle

elseif roombaY > k && switch\_dir == 0

if floorplan(roombaY, roombaX + 1) > 2

movedir = 2;

else

switch\_dir = 3;

movedir = 0;

end

%move left if obstacle up and right (switch directions)

elseif roombaY > k && floorplan(roombaY, roombaX - 1) > 2 && switch\_dir == 3

movedir = 4;

%move down if up, right, and left have obstacles

elseif roombaY > k && roombaX > m && floorplan(roombaY, roombaX - 1) < 3 && floorplan(roombaY + 1, roombaX) > 2

movedir = 3;

%------------------------------------------------------DOWN MOVEMENT-----------------------------------------------------

%move down if no obstacle

elseif roombaY < k && floorplan(roombaY + 1, roombaX) > 2

movedir = 3;

if switch\_dir == 4

switch\_dir = 0;

end

%move right if no obstacle is right and down is obstacle

elseif roombaY < k && switch\_dir == 0

if floorplan(roombaY, roombaX + 1) > 2

movedir = 2;

else

switch\_dir = 4;

movedir = 0;

end

%move left if obstacle down and right (switch directions)

elseif roombaY < k && floorplan(roombaY, roombaX - 1) > 2 && switch\_dir == 4

movedir = 4;

%move up if down, right, and left have obstacles

elseif roombaY < k && roombaX > m && floorplan(roombaY, roombaX - 1) < 3 && floorplan(roombaY - 1, roombaX) > 2

movedir = 1;

end

end