function [record] = main(map,person,max\_iter)

% %main

% load Personputter/personsLARGE

% load Maps/Building(1exit)LARGE

%max\_iter = 800;

record = recordinit(max\_iter,person);

for iter=1:max\_iter

person = action(person,map);

person = Forces(person,map);

record = recording(person,iter,record);

person = move(person,map);

end

end

function [record] = recordinit(max\_iter,person)

l = length(person.x);

record.time\_x = zeros(max\_iter,l);

record.time\_y = zeros(max\_iter,l);

record.time\_floor = zeros(max\_iter,l);

record.time\_force\_x = zeros(max\_iter,l);

record.time\_force\_y = zeros(max\_iter,l);

end

function person = action(person,map)

%takes persons out of the map, if they reach the exit

%moves them to another floor

%actions are defined by map.action

%1 means exit

%2 means change floor (to lower floor)

new\_person = person;

for i=1:length(person.x)

x = person.x(i);

y = person.y(i);

if map(person.level(i)).action(y,x) == 2 %change floor (put those cells generously around the stairs)

ind1 = find(person.level == person.level(i)-1); %take all persons on the lower floor

indx = find(person.x(ind1) == person.x(i)); %take all indices with the same x coords of ind1

indy = find(person.y(ind1) == person.y(i)); %take all indices with the same y coords of ind1

ind = intersect(indx,indy); %ind gives you the index of the person on the lower floor

%with the same coords as the current person

if isempty(ind)

new\_person.level(i) = person.level(i)-1; %change floor if there is no person blocking

end

end

end

function [person] = Forces(person,map)

a\_map = 0.3; %for LARGE 6, for normal 1

a\_pers = .1; %force parameter , for LARGE 1.5, for normal 0.75

for i=1:length(person.x)

floor = person.level(i);

%force by precomputed forcefield

person.force\_x(i)=map(floor).force\_x(person.y(i),person.x(i));

person.force\_y(i)=map(floor).force\_y(person.y(i),person.x(i));

%force by other persons

for k=1:length(person.x)

if i~=k && person.level(i)==person.level(k) %not itself, and only persons on the same floor

deltax = person.x(i)-person.x(k);

deltay = person.y(i)-person.y(k);

dist = (deltax^2+deltay^2);

if dist == 0

dist = 1;

end

force = a\_pers/dist;

person.force\_x(i) = person.force\_x(i) + force\*deltax/dist;

person.force\_y(i) = person.force\_y(i) + force\*deltay/dist;

end

end

end

function [record] = recording(person,iter,record)

l = length(person.x);

record.time\_x(iter,1:l) = person.x;

record.time\_y(iter,1:l) = person.y;

record.time\_floor(iter,1:l) = person.level;

record.time\_force\_x(iter,1:l) = person.force\_x;

record.time\_force\_y(iter,1:l) = person.force\_y;

end

function [person]=move(person,map)

[M N] = size(map(1).wall); %every map has the same size

for i=1:length(person.x)

x\_step = int32(person.force\_x(i));

y\_step = int32(person.force\_y(i));

x\_new = person.x(i) + x\_step;

y\_new = person.y(i) + y\_step;

if x\_new < N && x\_new > 0 && y\_new < M && y\_new > 0 %making sure it is inside the map

%making sure it isnt in the wall

x\_new1 = x\_new;

y\_new1 = y\_new;

if map(person.level(i)).wall(y\_new,x\_new) > 0

if map(person.level(i)).wall(y\_new,person.x(i)) == 0

x\_new1 = person.x(i);

y\_new1 = y\_new;

elseif map(person.level(i)).wall(person.y(i),x\_new) == 0

x\_new1 = x\_new;

y\_new1 = person.y(i);

end

end

person.x(i) = x\_new1;

person.y(i) = y\_new1;

end

%reset the forces

person.force\_x(i)=0;

person.force\_y(i)=0;

end

end

function visual(map,record,floor,nuller)

%visualizing stuff

figure(1)

set(1,'visible','off')

[M,N] = size(map(floor).wall);

x = [];

y = [];

for k=1:M

for l=1:N

if map(floor).wall(k,l) > 0

x = [x,l];

y = [y,k];

end

end

end

[numiter numpers] = size(record.time\_x);

for n = 1:numiter

hold on

%scatter(x,y,10,'k')

%scatter(x,y,10,'k','filled')

plot(x,y,'k.')

for m=1:numpers

if record.time\_floor(n,m) == floor

%hold on

%scatter(record.time\_x(n,m),record.time\_y(n,m),5,'r')

plot(record.time\_x(n,m),record.time\_y(n,m),'r.')

end

end

% xlim([0 N]);

% ylim([0 M]);

%pause(0.01);

%waitforbuttonpress();

disp(n);

%for saving the pictures

filename = 'C:\Users\joehla\'; %on alex' mac book pro

nuller = '10000'; %five letters

number = strcat(nuller(1:end-length(num2str(n))),num2str(n));

filename = strcat(filename,number);

saveas(gcf,filename,'jpg');

clf(gcf);

end

hold off

end

clear all

clc

[FileName,PathName] = uigetfile('\*.bmp', 'Select a Bitmap File');

I=imread(strcat(PathName,FileName));

[a b] = size(I);

for i=1:a

for j=1:b

if I(i,j)<50

I(i,j)=5;

end

if I(i,j)>200

I(i,j)=0;

end

if I(i,j)==140

I(i,j)=2;

end

if I(i,j)==115

I(i,j)=1;

end

end

end

% I(300,255)=2;

% I(200,150)=2;

% I(200,330)=2;

f = getFile\_my(I);

[FX,FY]=computeGradientField1(f);

function [F] = getFile\_my(I)

space=find(I==0);

exit=find(I==2);

passenger=find(I==1);

wall=find(I==5);

[n,m]=size(I);

F=zeros(n,m);

F(space)=1;

F(exit)=Inf;

F(passenger)=2;

F(wall)=0;

F=flipud(F);

end

%Personputter

numfloors = input('How many floors are there?');

if numfloors < 1

error('incorrect input')

end

person.x = [];

person.y = [];

person.level = [];

person.force\_x = [];

person.force\_y = [];

for k=1:numfloors

selection = 'Please Select floor number: ';

disp(strcat(selection,num2str(k)));

[FileName,PathName] = uigetfile('\*.bmp', 'Select the correct Bitmap File');

I=imread(strcat(PathName,FileName));

[y x] = find(I==140); %May be changed to other value if necessary

person.x = [person.x,x'];

person.y = [person.y,y'];

person.level = [person.level,k\*ones(1,length(x'))];

end

person.force\_x = zeros(1,length(person.x));

person.force\_y = zeros(1,length(person.x));

clear numfloors selection FileName PathName I x y k