Snake Game

Amr Mohamad Salah – 6287

Omar Alaa Abd-el-Kader – 6294

Mazen Mohamad Ibrahim – 6151

Nour El-din Hazem – 6261

Muhammad Ayman Fathy – 6144

Fares Ahmed Sobhy – 6337

-Game Introduction:

This is a version of the most popular game named “snake”. The main objective of this game is to feed an increasing length of a snake with food particles which are found at random positions.

You start the game by selecting the difficulty of the game out of four different difficulties, then you choose if you want boundaries or not.

Start moving by using the main four arrows (up, down, right and left) and grow your snake by collecting the RED PENTAGRAM shaped food and avoid touching the MAGENTA DIAMOND shape it’s a trap.

The game is over only if you touched the magenta diamond shape, hitting the boundaries or by the snake hitting its own body.

Notes:

-If no difficulty is chosen the difficulty is set to the default (which is the hardest mode).

-if boundary selection is skipped, no boundaries will appear.

-Press ‘Q’ to exit the game.

The Code:

(The comments are left intentionally)

```
function snake_game()
close all
%PRESS 'Q' TO EXIT GAME
%the magenta diamond shape is a trap BE CAREFUL!
%the red pentagram shape is your food
%Getting User Preferences
%If the user didn't choose an option for difficulty and
boundaries, its
%auto-set to the hardest difficulty and without boundaries
counter = 0;
difficulty = 0;
rate=0.047;
choice = menu('Choose
theDifficulty','easy','medium','hard','legendary');
```

```
if (choice==1)
   rate=0.433;
elseif (choice == 2)
   rate=0.233;
elseif (choice == 3)
   rate=0.1;
else
   rate=0.047;
end

boundchoice = menu('Do you want boundaries? ', 'Yes', 'No');
if (boundchoice == 1)
   bounds = 1;
else
   bounds = 0;
end
```

```
axis_limit= 15;  %Setting The Axis Limits
d=0;  %the direction of the snake
x =round(axis_limit/2);  %x coordinate starting point of snake
y =round(axis_limit/2); %y coordinate starting point of snake
d =randi([4]); % generates random direction to start in for snake
a =randi([axis_limit-1]); %generates random x coordinate for food
b =randi([axis_limit-1]); %generates random y coordinate for food
l=randi([axis_limit-1]); %generates random x coordinate for trap
k=randi([axis_limit-1]); %generates random y coordinate for trap
snake(1,1:2)=[x y]; %defines the snake for x and y coordinates
size_snake=1;
ate=0 ; %food ate by snake
ex=0;  %used to exit game
food=[a b];%defines food for a and b coordinates
trap=[l k];%defines trap for l and k coordinates
```

```
figure('KeyPressFcn',@my_callback);
   function my_callback(~,event)%callback function for
movement
       switch event.Character
          case 'q'
             ex=1;
          case 30              % arrow direction
             if(d~=2)
                d = 1;            %up d=1
             end
          case 31
             if(d~=1)
                d = 2;            %down d=2
             end
          case 29
             if(d~=4)
                d = 3;            %right d=3
             end
```

```
        case 28
            if(d~=3)
                d = 4;          %left d=4
            end
        end
    end

while (ex~=1)  %runs the snake as long as q is not pressed

%Making every row in snake matrix equal the previous row
%to show the snake movement.
    size_snake=size(snake);
    size_snake=size_snake(1);
    for l=size_snake+ate:-1:2
        snake(l,:)=snake(l-1,:);
    end
```

```
%generating the new coordinate for snake's head
   switch d        %calling callback function
      case 1
         snake(1,2)=snake(1,2)+1;  %add value of 1 to y position
      case 2
         snake(1,2)=snake(1,2)-1;  %subtract value of 1 to y
                                                position
      case 3
         snake(1,1)=snake(1,1)+1;%add value of 1 to x position
      case 4
         snake(1,1)=snake(1,1)-1;%subtracts value of 1 to x
                                                position
   end

  draw_snake(snake,food,size_snake,axis_limit, trap)
  %draws the snake difficulty makes game faster and
   %capping the maximum speed for pausing and resuming
  pause(max([rate .001]))
```

```
if snake(1,1)==food(1) && snake(1,2)==food(2)
        %if the snake head and food are in the same position
        ate=1;

        %Fixing the food appearing on the snake bug
        m = 1;
        f1 = randi([1 axis_limit-1]); %creates a new x position for
                                        the food
        f2 = randi([1 axis_limit-1]); %creates a new y position for
                                        the food
        while(size(snake, 1) ~= m)
        if ([f1 f2] == snake(m,:))

        f1 = randi([1 axis_limit-1]); %creates a new x position for
                                        the food
        f2 = randi([1 axis_limit-1]); %creates a new y position for
                                        the food
        end
        m = m + 1;
        end
```

```
m = 1;
t1 = randi([1 axis_limit-1]); %creates a new x position for
                                the trap
t2 = randi([1 axis_limit-1]); %creates a new y position for
                                the trap
 %this loop makes sure the trap doesn't get generated on the
%snake nor the food

while(size(snake, 1) ~= m)
if ([t1 t2] == snake(m,:))

t1 = randi([1 axis_limit-1]); %creates a new x position for
                                the trap
t2 = randi([1 axis_limit-1]); %creates a new y position for
                                the trap
end
m = m + 1;
end
```

```
    while([t1 t2] == [f1 f2])

    t1 = randi([1 axis_limit-1]);%creates a new x position for
                                 the trap
    t2 = randi([1 axis_limit-1]);%creates a new y position for
                                 the trap

    end

    food(1) = f1;
    food(2) = f2;
    trap(1) = t1;
    trap(2) = t2;

else
    ate=0;

end
```

```
% if the snake head touched the trap the player loses
    if snake(1,1)==trap(1) && snake(1,2)==trap(2)
        msgbox('You Lost! Try Again');
        ex=1;
    end


 %bounds;
 %Checks with if statement whether the snake head hit the
%boundaries or not


    if bounds==1
        %snake(1,:) %prints the coordinates in command window
                        (not i)
        if snake(1,1)==0  %if snake exceeds boundaries display
                                message box
            msgbox('YOU LOST! Try Again')
            ex=1;
        elseif snake(1,2)==0 %if snake exceeds boundaries display
                                    message box
            msgbox('YOU LOST! Try Again')
```

```
        ex=1;
    elseif snake(1,1)==axis_limit  %if snake exceeds
                                   boundaries display message box
        msgbox('YOU LOST! Try Again')
        ex=1;
    elseif snake(1,2)==axis_limit  %if snake exceeds
                                    boundaries display message box
        msgbox('YOU LOST! Try Again')
        ex=1;
    end
  else
  %in this line if a snake coordinate exceeds the limits then
%make the new coordinate for it is the original snake
%coordinate-axis limit+1 , in this case its (snake coordinate-16)
%this line is for upper and right boundaries(more than 15)
  % snake=snake-((snake>axis_limit).*(axis_limit+1));
  % so now the snake is negative so we make the
% new coordinate (the negative sign + the axis limits ) in this
%case its 16 so it appears on the other axis
```

```
        %this line is for left and down boundaries (less than 0)
        snake=snake+((snake<0).*(axis_limit+1));
    end

    %we use the sum method because plain if conditions
      crashes on starting
    %the game (SAME COORDINATES)

    %if snake hits itself
if (sum(snake(:, 1) ==snake(1, 1)   & snake(:, 2) == snake(1, 2)>1)
        msgbox('YOU LOST! Try Again')
         break
    end
  end
  close all
  end
function draw_snake(snake,food,size_snake,axis_limit, trap)
    for p = 1:size_snake
        plot(snake(p,1),snake(p,2), 'wo')
        hold on
```

```
    end
    plot(food(1,1),food(1,2), 'pr') %creates the vectors for the
                                    food and snake and plots them
    plot(trap(1,1),trap(1,2), 'md')
    whitebg([0 0 0])  %controls background color (black here)
    axis([0, axis_limit, 0, axis_limit])  %creates the axis for
                                          gameplay
    set(gcf, 'MenuBar', 'None')
    set(gca,'YTick',[])
    set(gca,'XTick',[])
    set(gcf,'Name','Snake Game','NumberTitle','off')
    hold off

end
```

-Problems that we faced:

1) The food appearing on the snake's body while moving → we solved this problem by adding if condition that generates a new random coordinates for the food if its coordinates is the same as the snake.

2) The pause function which is responsible for the speed of the snake → we made some calculations and tests to find the most suitable numbers for the speed.

3) The traps appearing on the snake's body or the food → just like the first problem we faced, we added another if condition to generate random coordinates for the trap if it's coordinates is the same as the food or the snake.