

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут»

**Лабораторна робота №3**  
*з дисципліни «Моделювання та оптимізація комп'ютерних  
систем»*

**«Моделювання клітинних автоматів»**

Виконав студент групи: КВ-11  
ПІБ: Терент'єв Іван Дмитрович

Перевірив: \_\_\_\_\_

**Київ 2024**

### **Загальне завдання**

1. Опрацювати теоретичну частину.
2. Ознайомитись із роботою програми *Life* в *MatLab*.
3. Вивчити роботу програми *Conway's life*.
4. Для «хаотичної» конфігурації, в якій кожна клітина знаходиться в стані 1 з імовірністю 50% розглянути часову еволюцію правил 00010010 (правила 18), правила 01001001 (правила 73) и правила 10001000 (правила 136).
5. Для "хаотичної" початкової конфігурації кожній клітині розглянути часову еволюцію правила Вольфрама з номером, який дорівнює номеру у списку групи помноженому на 5.

## ***Koд nporpaMy «Conway's life»***

% Conway's life with GUI

clf

clear all

%=====

% build the GUI

% define the plot button

plotbutton=uicontrol('style','pushbutton',...

    'string','Run',...

    'fontsize',12,...

    'position',[100,400,50,20],...

    'callback','run=1;');

% define the stop button

erasebutton=uicontrol('style','pushbutton',...

    'string','Stop',...

    'fontsize',12,...

    'position',[200,400,50,20],...

    'callback','freeze=1;');

% define the Quit button

quitbutton=uicontrol('style','pushbutton',...

    'string','Quit',...

    'fontsize',12,...

    'position',[300,400,50,20],...

    'callback','stop=1;close;');

```
number = uicontrol('style','text', ...  
    'string','1',...  
    'fontsize',12, ...  
    'position',[20,400,50,20]);
```

```
%=====
```

```
%CA setup
```

```
n=128;
```

```
%initialize the arrays
```

```
z = zeros(n,n);
```

```
cells = z;
```

```
sum = z;
```

```
%set a few cells to one
```

```
cells(n/2,.25*n:.75*n)=1;
```

```
cells(.25*n:.75*n,n/2)=1;
```

```
%cells(.5*n-1,.5*n-1)=1;
```

```
%cells(.5*n-2,.5*n-2)=1;
```

```
%cells(.5*n-3,.5*n-3)=1;
```

```
cells = (rand(n,n))<.5;
```

```
%how long for each case to stability or simple oscillators
```

```
%build an image and display it
```

```
imh = image(cat(3,cells,z,z));
```

```
set(imh, 'erasemode', 'none')
```

```
axis equal
```

axis tight

%index definition for cell update

x = 2:n-1;

y = 2:n-1;

%Main event loop

stop= 0; % wait for a quit button push

run = 0; % wait for a draw

freeze = 0; % wait for a freeze

while (stop==0)

if (run==1)

    %nearest neighbor sum

    sum(x,y)= cells(x,y-1) + cells(x,y+1) + ...

        cells(x-1, y) + cells(x+1,y) + ...

        cells(x-1,y-1) + cells(x-1,y+1) + ...

        cells(3:n,y-1) + cells(x+1,y+1);

    % The CA rule

    cells = (sum==3) | (sum==2 & cells);

    %draw the new image

    set(imh, 'cdata', cat(3,cells,z,z) )

    %update the step number display

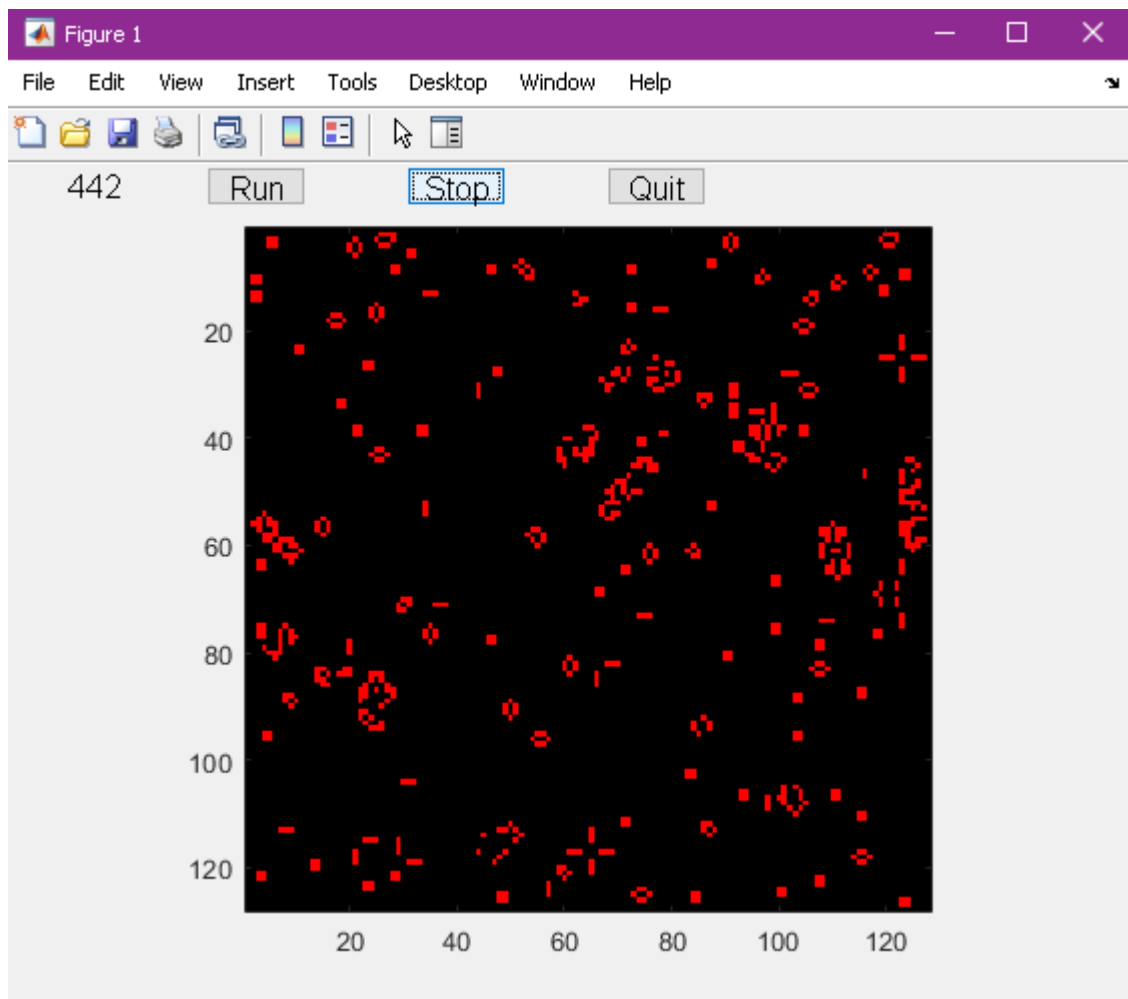
    stepnumber = 1 + str2num(get(number,'string'));

    set(number,'string',num2str(stepnumber))

end

if (freeze==1)

```
run = 0;
freeze = 0;
end
drawnow %need this in the loop for controls to work
end
```



*Приклад роботи програми Conway's life*

***Код програми для Rule 115(23\*5):***

%CODE136

clf

clear all

%=====

%build the GUI

%define the plot button

plotbutton=uicontrol('style','pushbutton',...

    'string','Run',...

    'fontsize',12,...

    'position',[100,400,50,20],...

    'callback','run=1;');

%define the stop button

erasebutton=uicontrol('style','pushbutton',...

    'string','Stop',...

    'fontsize',12,...

    'position',[200,400,50,20],...

    'callback','freeze=1;');

%define the Quit button

quitbutton=uicontrol('style','pushbutton',...

    'string','Quit',...

    'fontsize',12,...

    'position',[300,400,50,20],...

    'callback','stop=1;close;');

```
number = uicontrol('style','text', ...  
    'string','1', ...  
    'fontsize',12, ...  
    'position',[20,400,50,20]);
```

```
%=====
```

```
%CA setup
```

```
n=128;
```

```
%initialize the arrays
```

```
z = zeros(n,n);
```

```
cells = z;
```

```
sum = z;
```

```
%set a few cells to one
```

```
cells(n/2,.25*n:.75*n)=1;
```

```
cells(.25*n:.75*n,n/2)=1;
```

```
%cells(.5*n-1,.5*n-1)=1;
```

```
%cells(.5*n-2,.5*n-2)=1;
```

```
%cells(.5*n-3,.5*n-3)=1;
```

```
cells = (rand(n,n))<.5;
```

```
%how long for each case to stability or simple oscillators
```

```
%build an image and display it
```

```
imh = image(cat(3,cells,z,z));
```

```
set(imh, 'erasemode', 'none')
```

```
axis equal
```



axis tight

%index definition for cell update

%x = 2:n-1;

%y = 2:n-1;

x = 1;

y = 1;

%Main event loop

stop= 0; % wait for a quit button push

run = 0; % wait for a draw

freeze = 0; % wait for a freeze

while (stop==0)

if (run==1)

    %nearest neighbor sum

    x = x + 1;

    if(x == n+1)

        x = 1;

        y = y + 1;

        if(y == n+1)

            y = 1;

            x = 1;

        end

    end

    p = cells(x,y);

    r = cells(x,y);

    q = cells(x,y);

```

if(x-1 ~= 0)
    p = cells(x-1,y);
else
    if(y-1 ~= 0)
        p = cells(n,y-1);
    else
        p = 0;
    end
end

```

```

end
if(x+1 ~= n+1)
    r = cells(x+1,y);
else
    if(y+1 ~= n+1)
        r = cells(1,y+1);
    else
        r = 0;
    end
end
end

```

```

if(q == 1 && r == 1)
    cells(x,y)=1;
else
    cells(x,y)=0;
end

```

% The CA rule

```

%cells = (sum==3) | (sum==2 & cells);

```

%draw the new image

```

set(imh, 'cdata', cat(3,cells,z,z) )

```

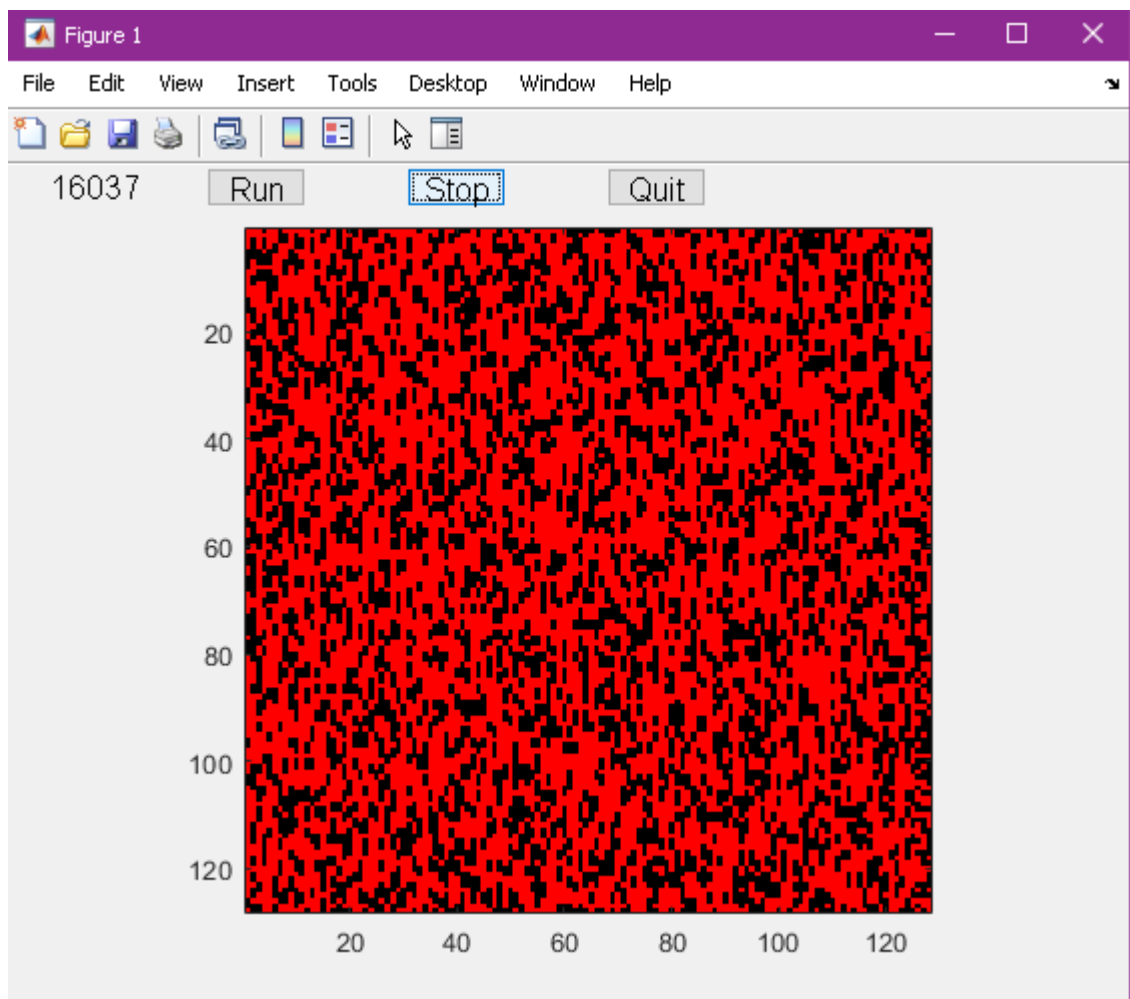
```

%update the step number display
stepnumber = 1 + str2num(get(number,'string'));
set(number,'string',num2str(stepnumber))
end

if (freeze==1)
    run = 0;
    freeze = 0;
end

drawnow %need this in the loop for controls to work
end

```



*Приклад роботи для Rule 115(23\*5)*

### ***Код програми для Rule 18***

```
%CODE18
```

```
clf
```

```
clear all
```

```
%=====
```

```
%build the GUI
```

```
%define the plot button
```

```
plotbutton=uicontrol('style','pushbutton',...
```

```
    'string','Run',...
```

```
    'fontsize',12,...
```

```
    'position',[100,400,50,20],...
```

```
    'callback','run=1;');
```

```
%define the stop button
```

```
erasebutton=uicontrol('style','pushbutton',...
```

```
    'string','Stop',...
```

```
    'fontsize',12,...
```

```
    'position',[200,400,50,20],...
```

```
    'callback','freeze=1;');
```

```
%define the Quit button
```

```
quitbutton=uicontrol('style','pushbutton',...
```

```
    'string','Quit',...
```

```
    'fontsize',12,...
```

```
    'position',[300,400,50,20],...
```

```
    'callback','stop=1;close;');
```

```
number = uicontrol('style','text', ...
```

```

    'string','1',...
    'fontsize',12, ...
    'position',[20,400,50,20]);

%=====

%CA setup

n=128;

%initialize the arrays
z = zeros(n,n);
cells = z;
sum = z;

%set a few cells to one
cells(n/2,25*n:.75*n)=1;
cells(.25*n:.75*n,n/2)=1;

%cells(.5*n-1,.5*n-1)=1;
%cells(.5*n-2,.5*n-2)=1;
%cells(.5*n-3,.5*n-3)=1;
cells = (rand(n,n))<.5;

%how long for each case to stability or simple oscillators

%build an image and display it
imh = image(cat(3,cells,z,z));
set(imh, 'erasemode', 'none')
axis equal
axis tight

```

```

%index definition for cell update

%x = 2:n-1;
%y = 2:n-1;
x = 1;
y = 1;

%Main event loop
stop= 0; % wait for a quit button push
run = 0; % wait for a draw
freeze = 0; % wait for a freeze

while (stop==0)

    if (run==1)
        %nearest neighbor sum
        x = x + 1;
        if(x == n+1)
            x = 1;
            y = y + 1;
            if(y == n+1)
                y = 1;
                x = 1;
            end
        end
        p = cells(x,y);
        r = cells(x,y);
        q = cells(x,y);
        if(x-1 ~= 0)

```

```

    p = cells(x-1,y);
else
    if(y-1 ~= 0)
        p = cells(n,y-1);
    else
        p = 0;
    end
end
if(x+1 ~= n+1)
    r = cells(x+1,y);
else
    if(y+1 ~= n+1)
        r = cells(1,y+1);
    else
        r = 0;
    end
end
end

if(q == 0)
    if(p == 1 && r == 1)
        cells(x,y) = 0;
    else
        if(p == 1 || r == 1)
            cells(x,y) = 1;
        end
    end
end
else
    cells(x,y) = 0;
end
end

```

```

% The CA rule

% cells = (sum==3) | (sum==2 & cells);

%draw the new image

set(imh, 'cdata', cat(3,cells,z,z) )

%update the step number diaplay

stepnumber = 1 + str2num(get(number,'string'));

set(number,'string',num2str(stepnumber))

end

if (freeze==1)

    run = 0;

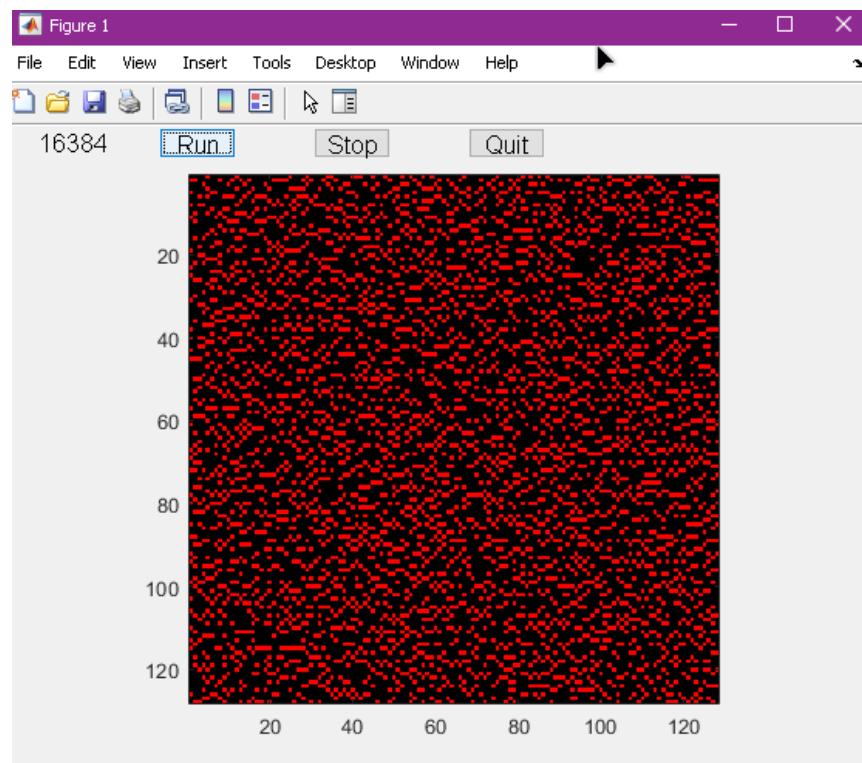
    freeze = 0;

end

drawnow %need this in the loop for controls to work

end

```



*Приклад роботи програми для Rule 18*



### ***Код програми для Rule 73***

%CODE73

clf

clear all

%=====

%build the GUI

%define the plot button

plotbutton=uicontrol('style','pushbutton',...

    'string','Run',...

    'fontsize',12,...

    'position',[100,400,50,20],...

    'callback','run=1;');

%define the stop button

erasebutton=uicontrol('style','pushbutton',...

    'string','Stop',...

    'fontsize',12,...

    'position',[200,400,50,20],...

    'callback','freeze=1;');

%define the Quit button

quitbutton=uicontrol('style','pushbutton',...

    'string','Quit',...

    'fontsize',12,...

    'position',[300,400,50,20],...

    'callback','stop=1;close;');

```
number = uicontrol('style','text', ...  
    'string','1', ...  
    'fontsize',12, ...  
    'position',[20,400,50,20]);
```

```
%=====
```

```
%CA setup
```

```
n=128;
```

```
%initialize the arrays
```

```
z = zeros(n,n);
```

```
cells = z;
```

```
sum = z;
```

```
%set a few cells to one
```

```
cells(n/2,.25*n:.75*n)= 1;
```

```
cells(.25*n:.75*n,n/2)= 1;
```

```
%cells(.5*n-1,.5*n-1)=1;
```

```
%cells(.5*n-2,.5*n-2)=1;
```

```
%cells(.5*n-3,.5*n-3)=1;
```

```
cells = (rand(n,n))<.5;
```

```
%how long for each case to stability or simple oscillators
```

```
%build an image and display it
```

```
imh = image(cat(3,cells,z,z));
```

```
set(imh, 'erasemode', 'none')
```

```
axis equal
```

axis tight

%index definition for cell update

%x = 2:n-1;

%y = 2:n-1;

x = 1;

y = 1;

%Main event loop

stop= 0; % wait for a quit button push

run = 0; % wait for a draw

freeze = 0; % wait for a freeze

while (stop==0)

if (run==1)

    %nearest neighbor sum

    x = x + 1;

    if(x == n+1)

        x = 1;

        y = y + 1;

        if(y == n+1)

            y = 1;

            x = 1;

        end

    end

    p = cells(x,y);

    r = cells(x,y);

    q = cells(x,y);

```

if(x-1 ~= 0)
    p = cells(x-1,y);
else
    if(y-1 ~= 0)
        p = cells(n,y-1);
    else
        p = 0;
    end
end
end
if(x+1 ~= n+1)
    r = cells(x+1,y);
else
    if(y+1 ~= n+1)
        r = cells(1,y+1);
    else
        r = 0;
    end
end
end

if(q == 0 && p == 0 && r == 0)
    cells(x,y) = 1;
else
    if(q == 1)
        if(p == 1 && r == 1)
            cells(x,y) = 0;
        else
            if(p == 1 || r == 1)
                cells(x,y) = 1;
            else

```

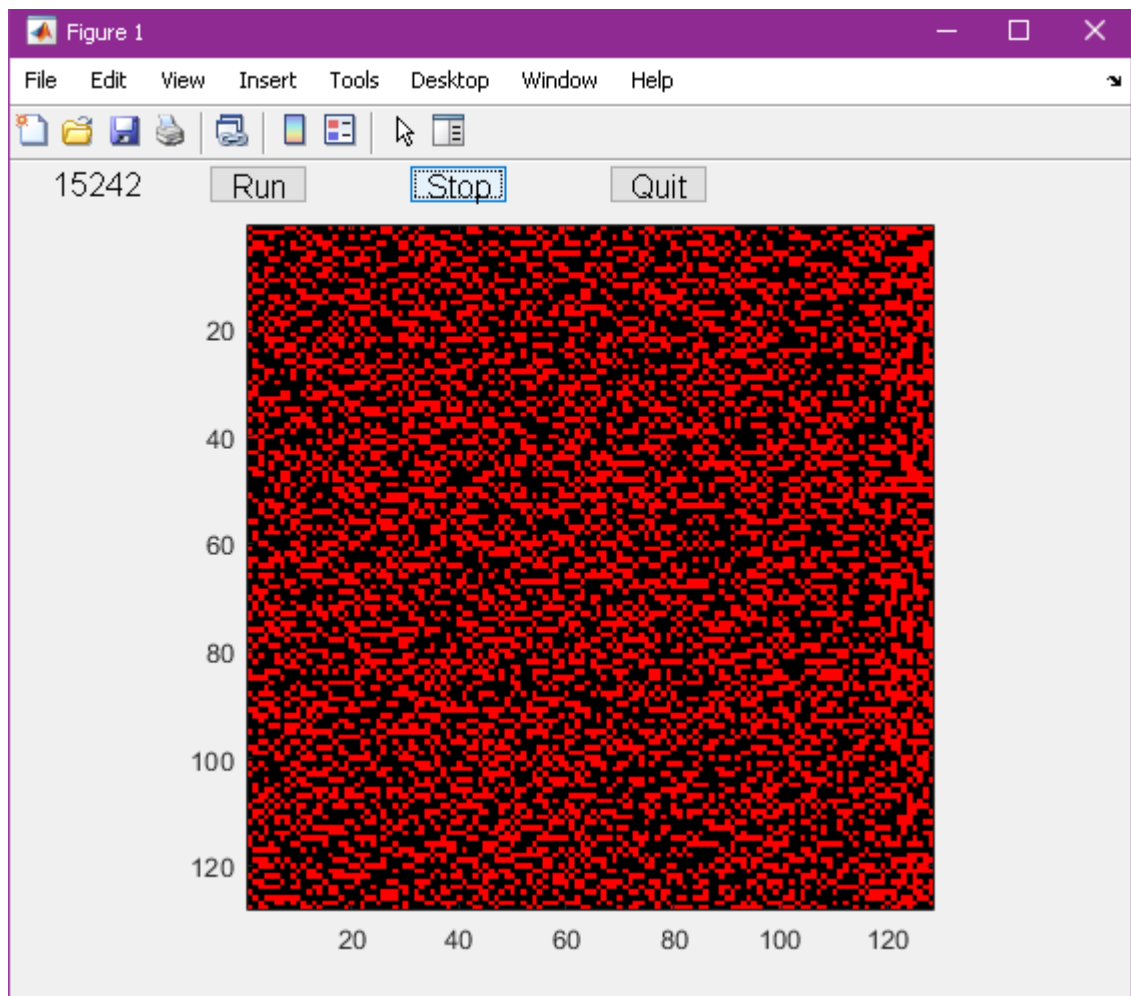
```

        cells(x,y) = 0;
    end
end
else
    cells(x,y) = 0;
end
end
% The CA rule
%cells = (sum==3) | (sum==2 & cells);

%draw the new image
set(imh, 'cdata', cat(3,cells,z,z) )
%update the step number display
stepnumber = 1 + str2num(get(number,'string'));
set(number,'string',num2str(stepnumber))
end

if (freeze==1)
    run = 0;
    freeze = 0;
end
drawnow %need this in the loop for controls to work
end

```



*Приклад роботи програми для Rule 73*

### ***Код програми для Rule 136***

```
%CODE136
```

```
clf
```

```
clear all
```

```
%=====
```

```
%build the GUI
```

```
%define the plot button
```

```
plotbutton=uicontrol('style','pushbutton',...
```

```
    'string','Run', ...
```

```
    'fontsize',12, ...
```

```
    'position',[100,400,50,20], ...
```

```
    'callback','run=1;');
```

```
%define the stop button
```

```
erasebutton=uicontrol('style','pushbutton',...
```

```
    'string','Stop', ...
```

```
    'fontsize',12, ...
```

```
    'position',[200,400,50,20], ...
```

```
    'callback','freeze=1;');
```

```
%define the Quit button
```

```
quitbutton=uicontrol('style','pushbutton',...
```

```
    'string','Quit', ...
```

```
    'fontsize',12, ...
```

```
    'position',[300,400,50,20], ...
```

```
    'callback','stop=1;close;');
```

```
number = uicontrol('style','text', ...
```

```

    'string','l',...
    'fontsize',12,...
    'position',[20,400,50,20]);

%=====

%CA setup

n=128;

%initialize the arrays
z = zeros(n,n);
cells = z;
sum = z;

%set a few cells to one
cells(n/2,.25*n:.75*n)= 1;
cells(.25*n:.75*n,n/2)= 1;

%cells(.5*n-1,.5*n-1)=1;
%cells(.5*n-2,.5*n-2)=1;
%cells(.5*n-3,.5*n-3)=1;
cells = (rand(n,n))<.5;

%how long for each case to stability or simple oscillators

%build an image and display it
imh = image(cat(3,cells,z,z));
set(imh,'erasemode','none')
axis equal
axis tight

```



```
%index definition for cell update
```

```
%x = 2:n-1;
```

```
%y = 2:n-1;
```

```
x = 1;
```

```
y = 1;
```

```
%Main event loop
```

```
stop= 0; % wait for a quit button push
```

```
run = 0; % wait for a draw
```

```
freeze = 0; % wait for a freeze
```

```
while (stop==0)
```

```
    if (run==1)
```

```
        %nearest neighbor sum
```

```
        x = x + 1;
```

```
        if(x == n+1)
```

```
            x = 1;
```

```
            y = y + 1;
```

```
            if(y == n+1)
```

```
                y = 1;
```

```
                x = 1;
```

```
            end
```

```
        end
```

```
        p = cells(x,y);
```

```
        r = cells(x,y);
```

```
        q = cells(x,y);
```

```
        if(x-1 ~= 0)
```

```

        p = cells(x-1,y);
    else
        if(y-1 ~= 0)
            p = cells(n,y-1);
        else
            p = 0;
        end
    end
end
if(x+1 ~= n+1)
    r = cells(x+1,y);
else
    if(y+1 ~= n+1)
        r = cells(1,y+1);
    else
        r = 0;
    end
end
end

if(q == 1 && r == 1)
    cells(x,y)=1;
else
    cells(x,y)=0;
end

% The CA rule
%cells = (sum==3) | (sum==2 & cells);

%draw the new image
set(imh, 'cdata', cat(3,cells,z,z))

%update the step number diaplay

```

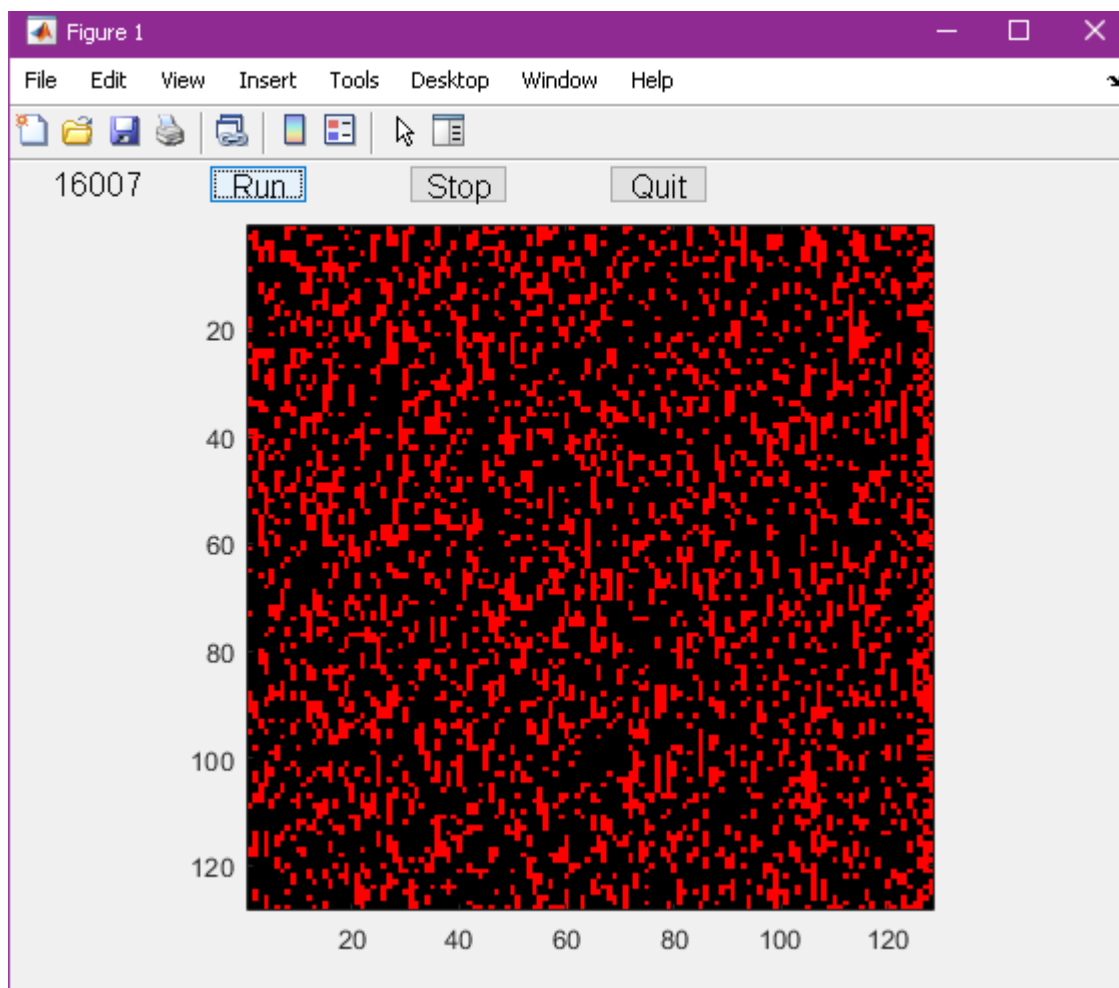
```

    stepnumber = 1 + str2num(get(number,'string'));
    set(number,'string',num2str(stepnumber))
end

if (freeze==1)
    run = 0;
    freeze = 0;
end

drawnow %need this in the loop for controls to work
end

```



*Приклад роботи програми для Rule 136*

**Правило 115(23\*5):** якщо поточний елемент білий, то поточний елемент буде чорним, якщо поточний елемент чорний та минулий чорний, то поточний залишиться чорним, інакше поточний елемент стане білим.

**Правило 18:** якщо поточний елемент білий, та наступний або минулий елементи чорні(але не обидва разом), то поточний елемент буде чорним, інакше білим.

**Правило 73:** якщо наступний, минулий та поточний елементи білі, то поточний елемент стане чорним, якщо поточний елемент чорний та наступний або минулий(але не обидва разом), то поточний елемент буде чорним, інакше білим.

**Правило 136:** якщо поточний та наступний елементи чорні, то елемент залишається чорним, інакше стає білим(значення минулого елементу неважливе).

***Висновок:***

Під час виконання лабораторної роботи була опрацьована теоретична частина, що стосувалася моделювання клітинних апаратів. Після чого ми ознайомилися з роботою програми Life в MatLab та вивчили роботу програми Conway's Life. Для хаотичної конфігурації, в якій кожна клітина знаходиться в стані 1 з імовірністю 50% була розглянута часова еволюція правил 18, 73, 136 та за варіантом 23 правило 115.