

```

clc
clear

% Display a welcome message
playedBefore = input('\nDo you want to play Connect 4? (0 for no, 1 for yes):
');

while(playedBefore ~= 0 && playedBefore ~= 1)
    errordlg('Invalid input, please enter 0 or 1!\n');
    playedBefore = input('Do you want to play Connect 4? (0 for no, 1 for
yes): ');
end

if(playedBefore == 1)
    msgbox('Welcome! Nice to meet you. ');
    player_1 = input('Player 1 (use red chip, move first), please enter your
name:', 's');
    player_2 = input('Player 2 (use black chip), please enter your
name:', 's');
else
    msgbox('Ok. As you wish. Goodbye!');
    return;
end

% Initialize scene
my_scene = simpleGameEngine('ConnectFour.png',86,101);

% Set up variables to name the various sprites
empty_sprite = 1;
red_sprite = 2;
black_sprite = 3;

% Display empty board
board_display = empty_sprite * ones(6,7);
drawScene(my_scene,board_display)

% Start the game
color = 0; % To determine the color of a chip player, assign 1 to red and
assign 0 to black
winner = 0; % when winner = 1, there is a winner or the board is full(even).
[r,c] = getMouseInput(my_scene);

% The first input should be in last/bottom row
while r ~= 6
    errordlg('Please choose the row 6 for the first turn!');
    pause(1.5);
    [r,c] = getMouseInput(my_scene);
end
board_display(r,c) = red_sprite;
drawScene(my_scene,board_display)

while winner == 0
    % Start to drop the chip
    if color == 1
        [r,c] = getMouseInput(my_scene);
    end
end

```

```

% checking if a spot is already filled
while board_display(r,c) ~= empty_sprite
    errordlg('Please click an empty sprite');
    pause(1.5);
    [r,c] = getMouseInput(my_scene);
end
    count = sum(board_display(:,c)~=1);

    % determining the lowest tile
    switch 6 - r - count
        case 0
            r;
        case 1
            r = r + 1;
        case 2
            r = r + 2;
        case 3
            r = r + 3;
        case 4
            r = r + 4;
        case 5
            r = r + 5;
    end
    board_display(r,c) = red_sprite;
    drawScene(my_scene,board_display)
color = 0;
% check each column, each row and each diagonal
% winning conditions for each player
for m = 1:6
    for n = 1:4
        if board_display(m,(n:n+3)) == 2
            msgbox('player 1 wins! Thanks for playing!')
            pause(1.5);
            winner = 1;
        end
    end
end
for m = 1:3
    for n = 1:7
        if board_display((m:m+3),n) == 2
            msgbox('player 1 wins! Thanks for playing!')
            pause(1.5);
            winner = 1;
        end
    end
end

% diagonal
for m = 1:3
    for n = 1:4
        diagonalA =
[board_display(m,n),board_display(m+1,n+1),board_display(m+2,n+2),board_displ
ay(m+3,n+3)];
        if unique(diagonalA) == 2
            msgbox('player 1 wins! Thanks for playing!')

```

```

        pause(1.5);
        winner = 1;
    end
end
end
for m = 4:6
    for n = 1:4
        diagonalB = [board_display(m,n),board_display(m-
1,n+1),board_display(m-2,n+2),board_display(m-3,n+3)];
        if unique(diagonalB) == 2
            msgbox('player 1 wins! Thanks for playing!')
            pause(1.5);
            winner = 1;
        end
    end
end
else

    % player 2
    [r,c] = getMouseInput(my_scene);
    while board_display(r,c) ~= empty_sprite
        errordlg('Please click an empty sprite!');
        pause(1.5);
        [r,c] = getMouseInput(my_scene);
    end
    count = sum(board_display(:,c)~=1);
    switch 6 - r - count
        case 0
            r;
        case 1
            r = r + 1;
        case 2
            r = r + 2;
        case 3
            r = r + 3;
        case 4
            r = r + 4;
        case 5
            r = r + 5;
    end
    board_display(r,c) = black_sprite;
    drawScene(my_scene,board_display)
color = 1;
% check each column, each row and each diagonal
for m = 1:6
    for n = 1:4
        if board_display(m,(n:n+3)) == 3
            msgbox('player 2 wins! Thanks for playing!')
            pause(1.5);
            winner = 1;
        end
    end
end
for m = 1:3
    for n = 1:7
        if board_display((m:m+3),n) == 3
            msgbox('player 2 wins! Thanks for playing!')

```

```

        pause(1.5);
        winner = 1;
    end
end
end
for m = 1:3
    for n = 1:4
        diagonalA =
[board_display(m,n),board_display(m+1,n+1),board_display(m+2,n+2),board_displ
ay(m+3,n+3)];
        if unique(diagonalA) == 3
            msgbox('player 2 wins! Thanks for playing!')
            pause(1.5);
            winner = 1;
        end
    end
end
for m = 4:6
    for n = 1:4
        diagonalB = [board_display(m,n),board_display(m-
1,n+1),board_display(m-2,n+2),board_display(m-3,n+3)];
        if unique(diagonalB) == 3
            msgbox('player 2 wins! Thanks for playing!')
            pause(1.5);
            winner = 1;
        end
    end
end
end
end

% Determining the tie situation
if board_display(1:6,1:7) ~= 1
    msgbox('You guys are even. Let us play one more time!')
    pause(1.5);
    winner = 1;
end
end
end

```