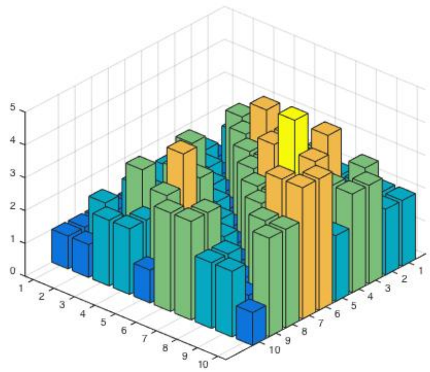
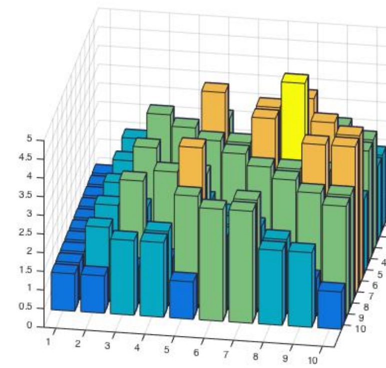
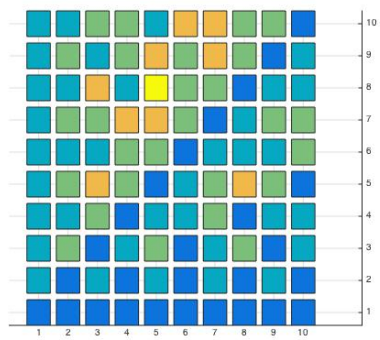
1. For any input pair in which ***m*** is smaller than ***n*** (***0 <= m < n***), the Euclid Algorithm Simply swaps the two numbers for the first iteration. This swap happens only one time since ***gcd(n, m*mod(n))*** implies that each in each successive iteration of the algorithm the first number is greater than the second number.
2.







```
function [ ans ] = Euclidean( a, b )
    count=0; % initialize counter
    a=abs(a);
    b=abs(b);
    if(a<b)  % switch a and b, if necessary, so that b<a
        c=a; % hang onto the value of a
        a=b; % even while replacing a with b
        b=c; % now replace b with a
        count=count+1;
    end
    while(b>0)
        q=floor(a/b);
        r=a-q*b;
        a=b;
        b=r;
        count=count+1;
    end
    ans = [ count ];
end

figure
matrix = zeros(10,10);
for n = 1:10
    for m = 1:10
        matrix(n,m) = Euclidean(n,m)
    end
end

h = bar3(matrix, 'detached');
for i = 1:length(h)
    zdata = get(h(i),'Zdata');
    set(h(i),'Cdata',zdata)
end
```

*2a.* As we can see in the graph the minimum number of divisions is 1.

*2b.* As we se from the graph the maximum number of division is 5, when it is computed the gdc(5,8).

*3.* *a.*

```
function [ ans ] = EuclideanSubtraction( a, b )
    count=0; % initialize counter
    a=abs(a);
    b=abs(b);
    while(b~=0)
    if(a<b)  % switch a and b, if necessary, so that b<a
        c=a; % hang onto the value of a
        a=b; % even while replacing a with b
        b=c; % now replace b with a
    end
    r=a-b;
    a=b;
    b=r;
    count=count+1;
    end
    ans = [ a ];
end
```

*b.* From the Euclide's game we can see that the number you can use in the game are the integers created from the Euclid's Subtraction Algorithm.(Showed Above) Even if the numbers are appearing in different others in the board, at the end of the game all numbers have to appear. Therefore, if *m/gdc(m,n)*, where m is the maximum number of the two appearing at the beginning, is even you let the other player to go first, if it is odd you go first.

4.

```
function [ sqr ] = findSqrt( x )
    sqr = 0;
    for n = 1:x
        if(n^2 <= x)
            sqr = sqr + 1;
        else
            break;
        end
    end
end
```

5.

```
list1 = [2, 5, 5, 5];
list2 = [2, 2, 3, 5, 5, 7];
result = [];

l1 = length(list1);
l2 = length(list2);


for i = 1:l1
    for j = 1:l2
        if(list1(i) == list2(j))
            if not(any(list1(i)==result))
                result = cat(1, result, list1(i));
            end
        end
    end
end
```