Cmpe 360 CG
Project 2

Neslihan Pelin Metin
71047171244

Part 1

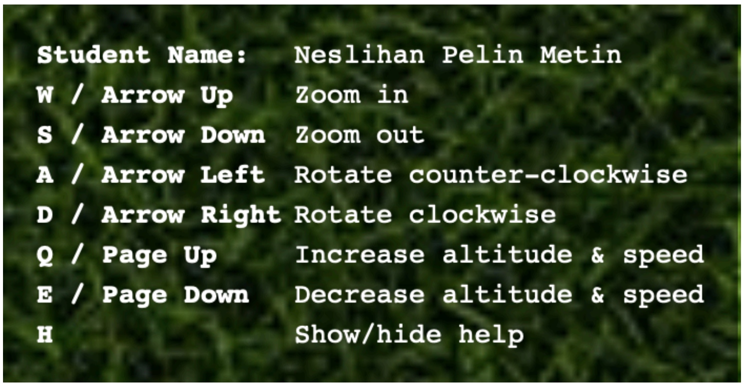

Image 0: The help bar that shows us how to use the drone.


Image 1: The page that appears when we open the html file.

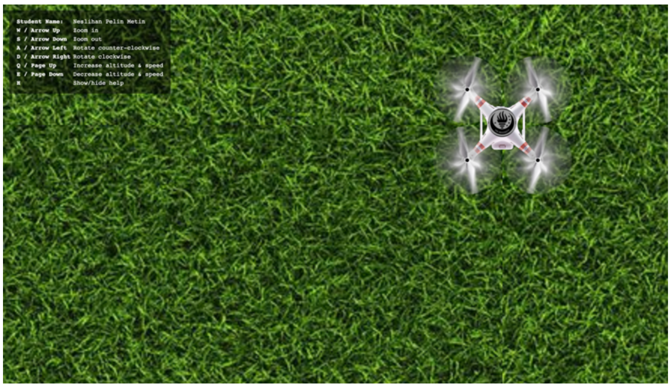
Image 2: This is how it looks when we move the mouse.


Image 3: When we press the W key, it zooms into the drone.

Neslihan Pelin Metin
71047171244

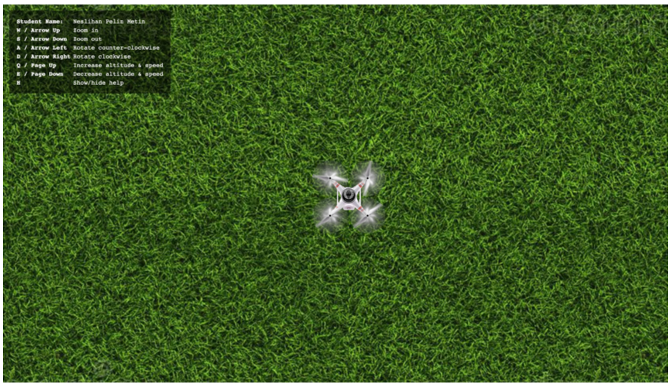

Image 4: When we press the S key, it moves away from the drone.

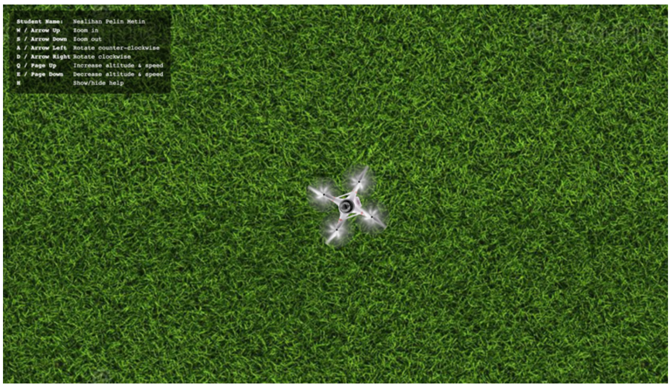

Image 5: When we press the A key, the drone turns counterclockwise.

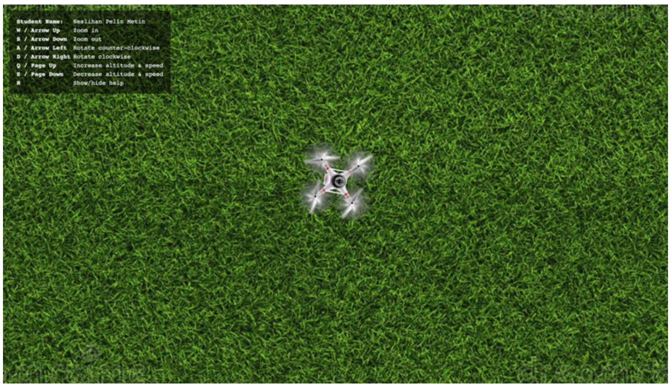

Image 6: When we press the A key, the drone turns clockwise.

Image 7: When we press the letter Q, the drone increases its speed and moves in the direction of the torch tip of the Ted University logo.

Neslihan Pelin Metin
71047171244

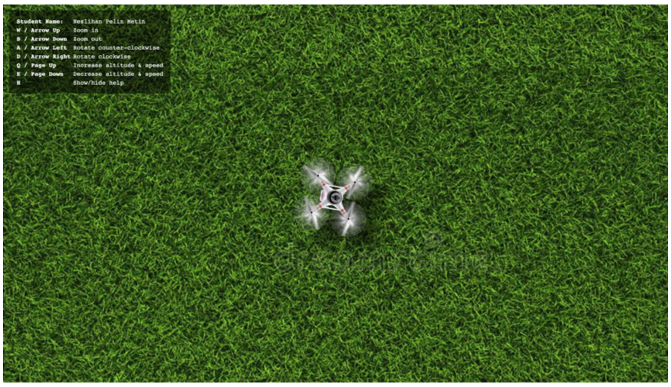

Image 8: When we press the letter E, the drone decreases its speed.

Recording Drive Link:
https://drive.google.com/file/d/15zIcv5GvrI3QKblvtorXr7buMWfxGO1Y/view

Part 2

1. function GetTransform(positionX, positionY, rotation, scale)

```
function GetTransform(positionX, positionY, rotation, scale) {

    var cos = Math.cos((rotation * Math.PI) / 180);
    var sin = Math.sin((rotation * Math.PI) / 180);

    var c = cos * scale;
    var s = sin * scale;

    var matrix = [c, s, 0, -s, c, 0, positionX, positionY, 1];

    return matrix; }
```

In this function we had to get the 3x3 matrix which will perform scaling, rotation, and translation respectively. I reached the matrix by following the operations below.

$$\begin{bmatrix} 1 & 0 & x_t \\ 0 & 1 & y_t \\ 0 & 0 & 1 \end{bmatrix} \times \left(\underbrace{\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}} \right)$$

$$\begin{bmatrix} 1 & 0 & x_t \\ 0 & 1 & y_t \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} s_x\cos\phi & -s_y\sin\phi & 0 \\ s_x\sin\phi & s_y\cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x\cos\phi & -s_y\sin\phi & x_t \\ s_x\sin\phi & s_y\cos\phi & y_t \\ 0 & 0 & 1 \end{bmatrix}$$

$s_x = s_y = scale$
$x_t = positionX$
$y_t = positionY$

$\phi = (rotation \cdot Math.PI) / 180$
$c = \cos\phi \cdot scale$
$s = \sin\phi \cdot scale$

$$\begin{bmatrix} c & -s & positionX \\ s & c & positionY \\ 0 & 0 & 1 \end{bmatrix}$$

I multiplied the rotation and scaling matrices first. Then I multiplied the translation matrix with the matrix I have found before. Finally, I simply put my calculations into the code and after that, I returned the resulting matrix in the specified format.

Neslihan Pelin Metin
71047171244

$$\begin{pmatrix} array[0] & array[3] & array[6] \\ array[1] & array[4] & array[7] \\ array[2] & array[5] & array[8] \end{pmatrix}$$

2. function ApplyTransform(trans1, trans2)

```
function ApplyTransform(trans1, trans2) {
    var result = new Array(9);

    for(var i = 0; i < 9; i++){
        result[i] = 0;
    }

    for(var i = 0; i < 3; i++)
    {
        for(var j = 0; j < 3; j++)
        {
            for(k = 0; k < 3; k++)
            {
                result[3*i+j] = result[3*i+j] + trans1[3*i+k] * trans2[3*k+j];
            }
        }
    }

    return result; }
```

In this function, we were asked to find the multiplication of matrices. To be able to do this, I have used a matrix multiplication algorithm that I found on an article. (Syed M. Qasim, Hardware Realization of Matrix Multiplication using Field Programmable Gate Array, 2009)[1]

```
procedure MatrixMultiplication(A, B)
  input A, B n*n matrix
  output C, n*n matrix

begin
  for ( i = 0; i < n; i++)
    for ( j = 0; j < n; j++)
      C[i,j] = 0;
    end for
  end for

  for ( i = 0; i < n; i++)
    for ( j = 0; j < n; j++)
      for( k = 0; k < n; k++)
       C[i,j] = C[i,j] + A[i,k] * B[k,j]
      end for
    end for
  end for
end MatrixMultiplication
```

In this pseudocode the matrices were stored in 2D arrays, so I changed the indexes of the matrices to make it stored as 1D.

Reference:
https://www.researchgate.net/publication/258926922_Hardware_Realization_of_Matrix_Multiplication_using_Field_Programmable_Gate_Array