TIC TAC TOE PROJECT REPORT

Tools Used:

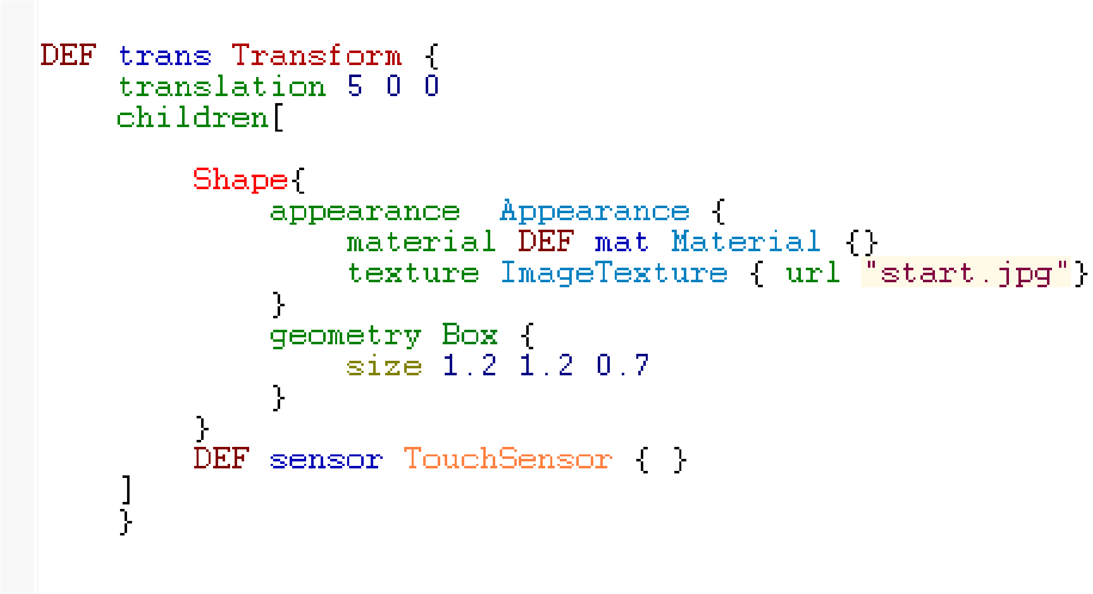
- VRML pad as an editor
- Cortona 3D viewer as a viewer for VRML files
- Cosmo Player as a VRML plugin
- Online Audio converter: http://audio.online-convert.com/convert-to-wav

Project Design:

- The grid of the game is made up of 9 blocks put together in a 3X3 grid
- A block was created using an IndexedFaceSet and covered by an image that has an "O" on top and an "X" at the bottom
- The IndexedFaceSet was then translated to each of the positions in the grid.
- On clicking of any of the blocks, the box rotates depending on whose turn it is and an "X" or "O" appears in the block.
- If a player places 3 Xs or Os in a row, he wins, a sound is played, and a message displaying the winning player appears. Otherwise, the message displayed is "Draw!"
- A "New Game" button was added and translated to the right of the grid to enable the players to clear the board and start a new game.

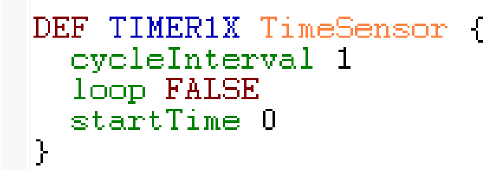
Code Snippets:

This is the creation of the “New Game” button:



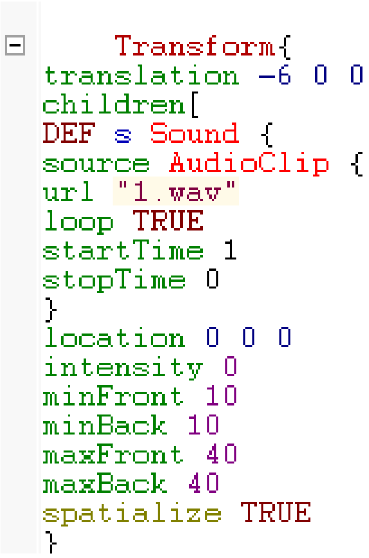
```
DEF trans Transform {
     translation 5 0 0
     children[

          Shape{
               appearance  Appearance {
                    material DEF mat Material {}
                    texture ImageTexture { url "start.jpg"}
               }
               geometry Box {
                    size 1.2 1.2 0.7
               }
          }
          DEF sensor TouchSensor { }
     ]
     }
```

This is the time sensor. Two were created for each button:



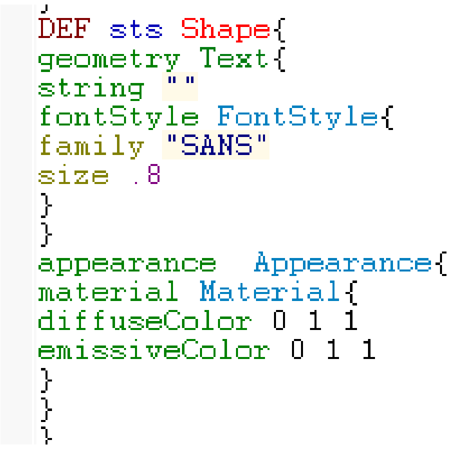
```
DEF TIMER1X TimeSensor {
  cycleInterval 1
  loop FALSE
  startTime 0
}
```

This adds the soundtrack to the game:



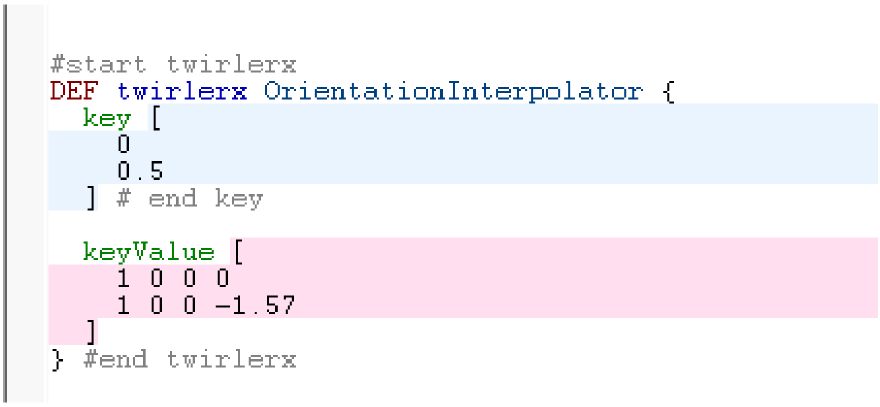
```
    Transform{
translation -6 0 0
children[
DEF s Sound {
source AudioClip {
url "1.wav"
loop TRUE
startTime 1
stopTime 0
}
location 0 0 0
intensity 0
minFront 10
minBack 10
maxFront 40
maxBack 40
spatialize TRUE
}
```

This is the part where we add a string to display who whens or whether it’s a draw:



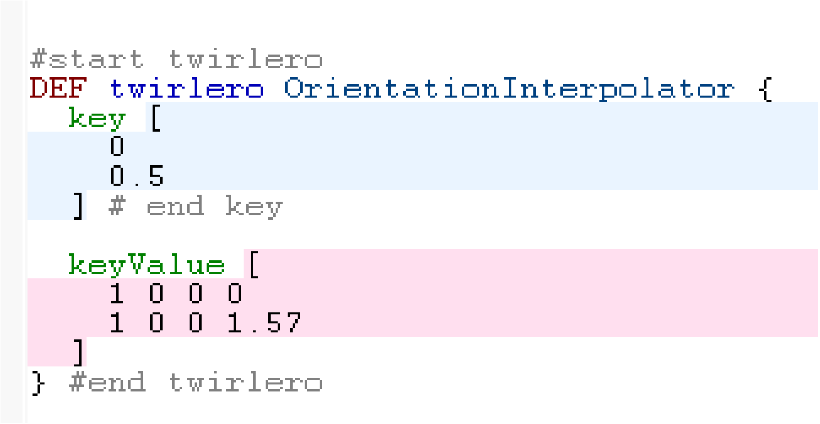
```
}
DEF sts Shape{
geometry Text{
string ""
fontStyle FontStyle{
family "SANS"
size .8
}
}
appearance  Appearance{
material Material{
diffuseColor 0 1 1
emissiveColor 0 1 1
}
}
}
```

This is the OrientationInterpolator that is responsible for the rotation of the shape when an “X” is placed in the box. One is defined for each box



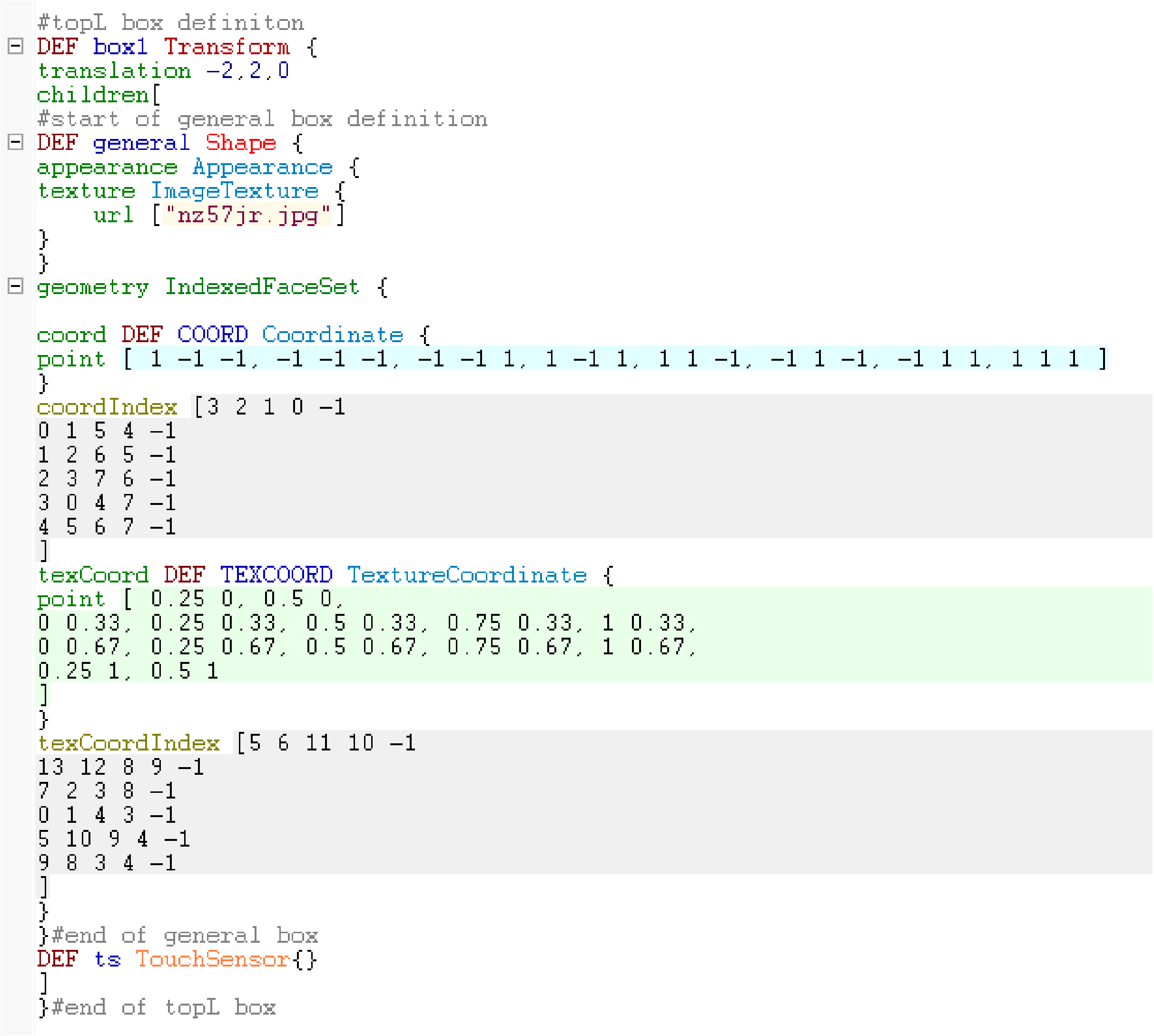
```
#start twirlerx
DEF twirlerx OrientationInterpolator {
   key [
      0
      0.5
   ] # end key

   keyValue [
      1 0 0 0
      1 0 0 -1.57
   ]
} #end twirlerx
```

This is the OrientationInterpolator that is responsible for the rotation of the shape when an “O” is placed in the box. One is also defined for each box



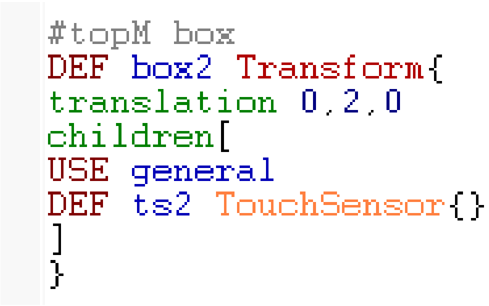
```
#start twirlero
DEF twirlero OrientationInterpolator {
   key [
      0
      0.5
   ] # end key

   keyValue [
      1 0 0 0
      1 0 0 1.57
   ]
} #end twirlero
```

This is the creation of the box. A touch sensor is added to detect the clicking of the box. It is translated to the coordinates -2, 2, 0. This is the top left box



```
#topL box definiton
DEF box1 Transform {
translation -2,2,0
children[
#start of general box definition
DEF general Shape {
appearance Appearance {
texture ImageTexture {
    url ["nz57jr.jpg"]
}
}
geometry IndexedFaceSet {

coord DEF COORD Coordinate {
point [ 1 -1 -1, -1 -1 -1, -1 -1 1, 1 -1 1, 1 1 -1, -1 1 -1, -1 1 1, 1 1 1 ]
}
coordIndex [3 2 1 0 -1
0 1 5 4 -1
1 2 6 5 -1
2 3 7 6 -1
3 0 4 7 -1
4 5 6 7 -1
]
texCoord DEF TEXCOORD TextureCoordinate {
point [ 0.25 0, 0.5 0,
0 0.33, 0.25 0.33, 0.5 0.33, 0.75 0.33, 1 0.33,
0 0.67, 0.25 0.67, 0.5 0.67, 0.75 0.67, 1 0.67,
0.25 1, 0.5 1
]
}
texCoordIndex [5 6 11 10 -1
13 12 8 9 -1
7 2 3 8 -1
0 1 4 3 -1
5 10 9 4 -1
9 8 3 4 -1
]
}
}#end of general box
DEF ts TouchSensor{}
]
}#end of topL box
```

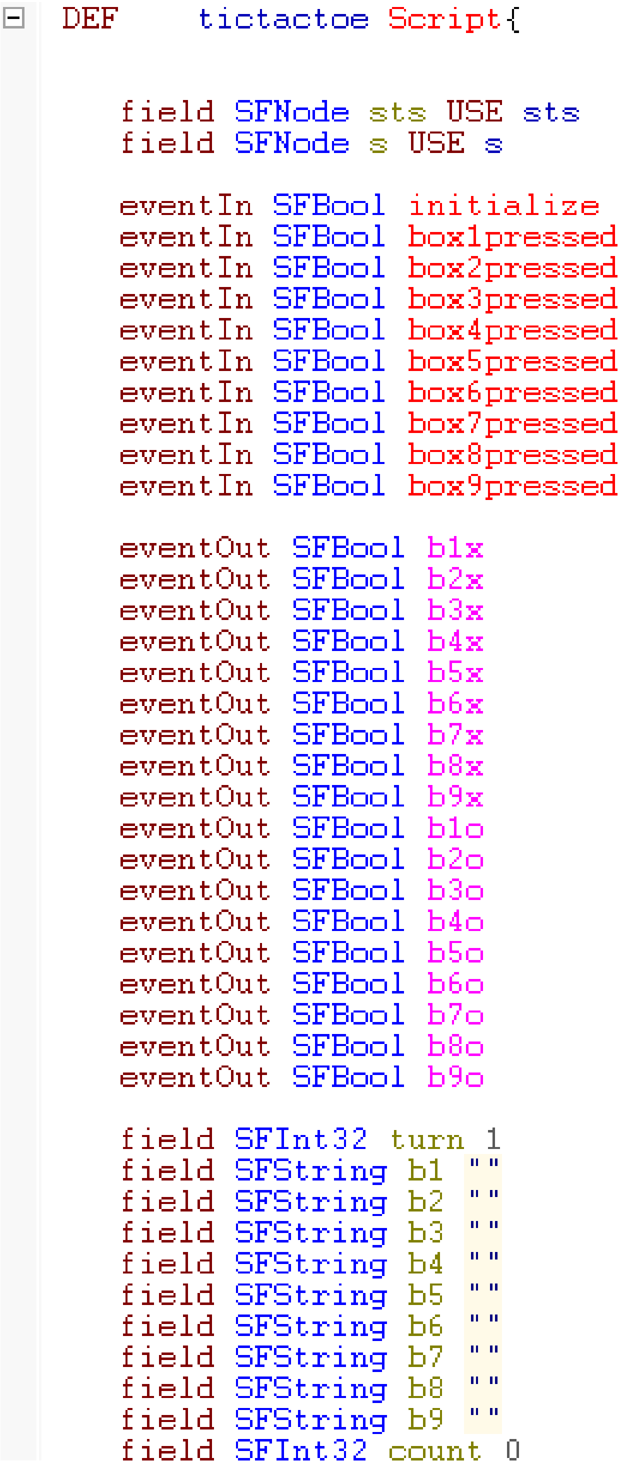
The USE statement is used to paste the same box 8 more times to complete the grid. A separate sensor is added to each box



```
#topM box
DEF box2 Transform{
translation 0,2,0
children[
USE general
DEF ts2 TouchSensor{}
]
}
```

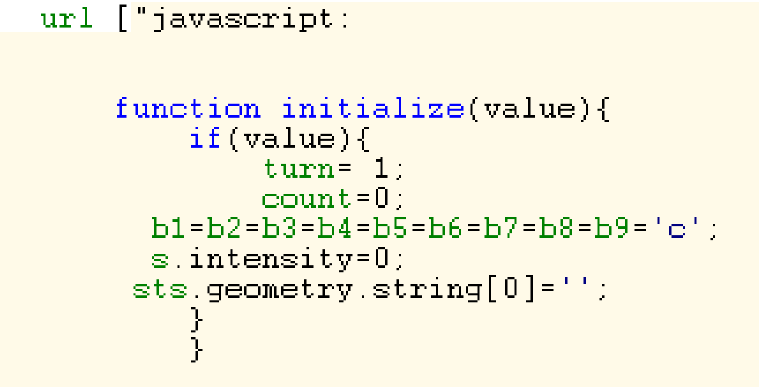
This is the start of the script, with all the definitions:

The Script is called tictactoe. It has two SFNodes called sts and s. s is responsible for the winning sound, and sts for the string that displays the winning player. It has 10 eventIns if type SFBool. 9 of those are to connect to the sensors of the 9 boxes, and one to initialize the boxes and clear them. There are 18 eventOuts of tye SFBool, these indicate whether an X or O has been pressed. The SFStrings and SFInts are initializations to strings that are used later on.



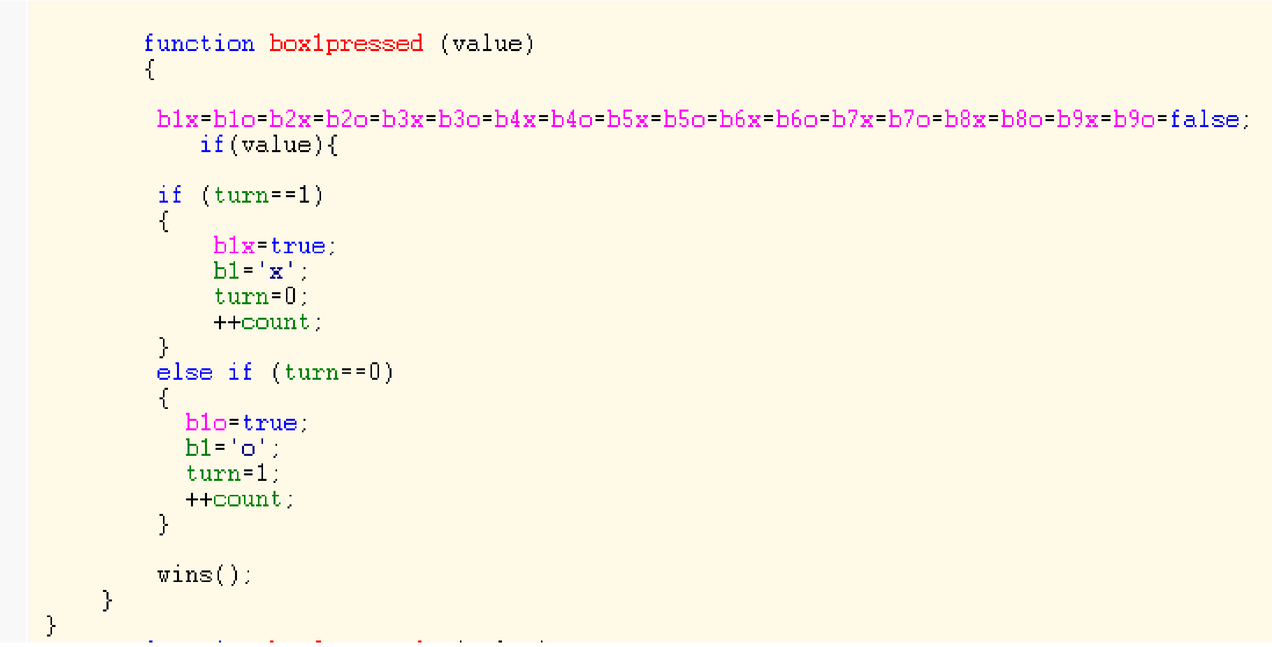
```
DEF    tictactoe Script{


    field SFNode sts USE sts
    field SFNode s USE s

    eventIn SFBool initialize
    eventIn SFBool box1pressed
    eventIn SFBool box2pressed
    eventIn SFBool box3pressed
    eventIn SFBool box4pressed
    eventIn SFBool box5pressed
    eventIn SFBool box6pressed
    eventIn SFBool box7pressed
    eventIn SFBool box8pressed
    eventIn SFBool box9pressed

    eventOut SFBool b1x
    eventOut SFBool b2x
    eventOut SFBool b3x
    eventOut SFBool b4x
    eventOut SFBool b5x
    eventOut SFBool b6x
    eventOut SFBool b7x
    eventOut SFBool b8x
    eventOut SFBool b9x
    eventOut SFBool b1o
    eventOut SFBool b2o
    eventOut SFBool b3o
    eventOut SFBool b4o
    eventOut SFBool b5o
    eventOut SFBool b6o
    eventOut SFBool b7o
    eventOut SFBool b8o
    eventOut SFBool b9o

    field SFInt32 turn 1
    field SFString b1  ""
    field SFString b2  ""
    field SFString b3  ""
    field SFString b4  ""
    field SFString b5  ""
    field SFString b6  ""
    field SFString b7  ""
    field SFString b8  ""
    field SFString b9  ""
    field SFInt32 count 0
```

This is the start of the JavaScript section. The function initialize is connected to the eventIn “initialize”:



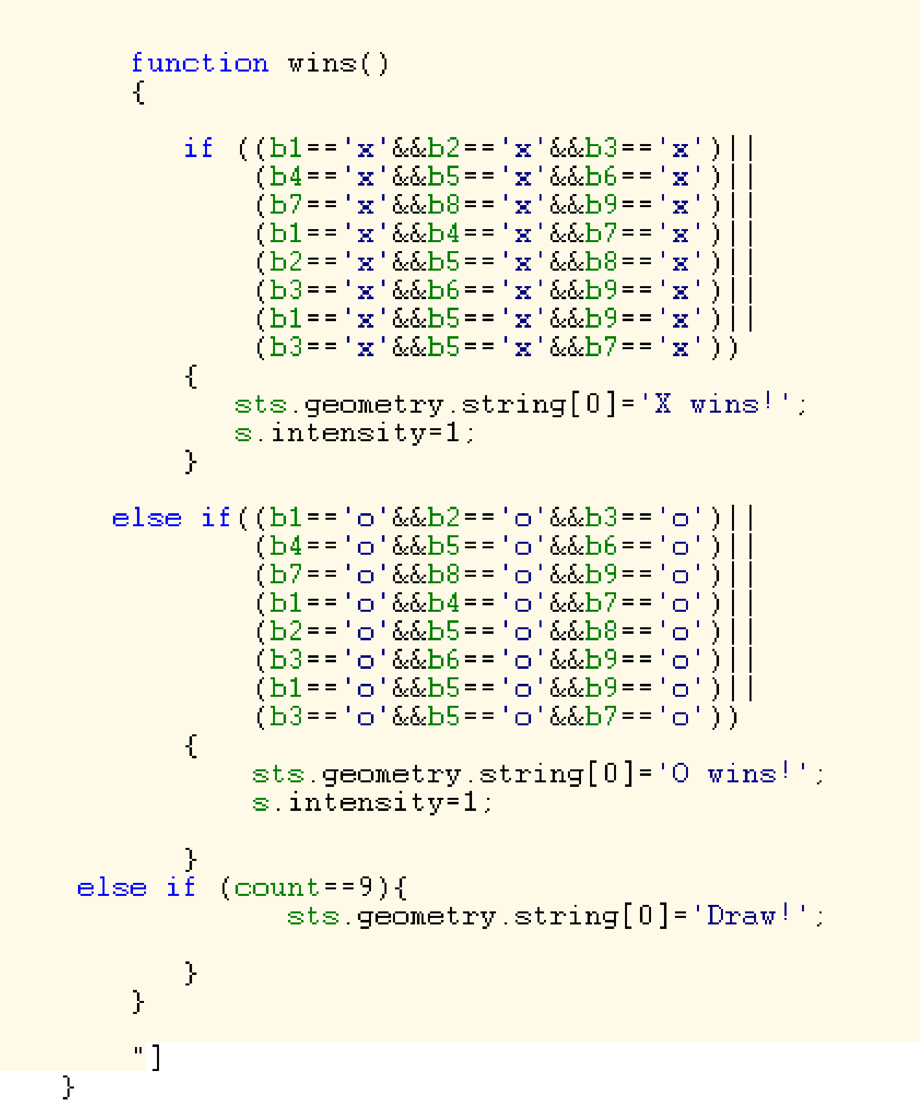
```
url ["javascript:

    function initialize(value){
        if(value){
            turn= 1;
            count=0;
     b1=b2=b3=b4=b5=b6=b7=b8=b9='c';
     s.intensity=0;
    sts.geometry.string[0]='';
        }
        }
```

This is the function that handles if a box is pressed. There are 9 of this, one for each box:



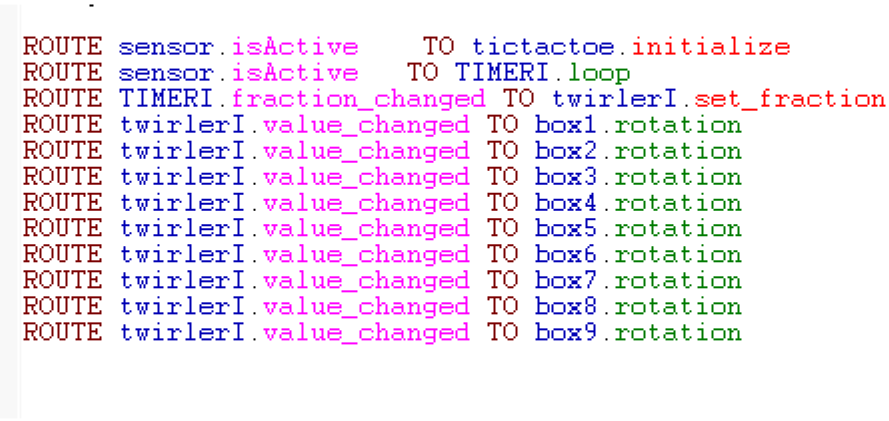
```
        function box1pressed (value)
        {

         b1x=b1o=b2x=b2o=b3x=b3o=b4x=b4o=b5x=b5o=b6x=b6o=b7x=b7o=b8x=b8o=b9x=b9o=false;
            if(value){

         if (turn==1)
         {
             b1x=true;
             b1='x';
             turn=0;
             ++count;
         }
         else if (turn==0)
         {
           b1o=true;
           b1='o';
           turn=1;
           ++count;
         }

         wins();
    }
}
```

The function wins is called to determine if a player wins, or if it is a draw:

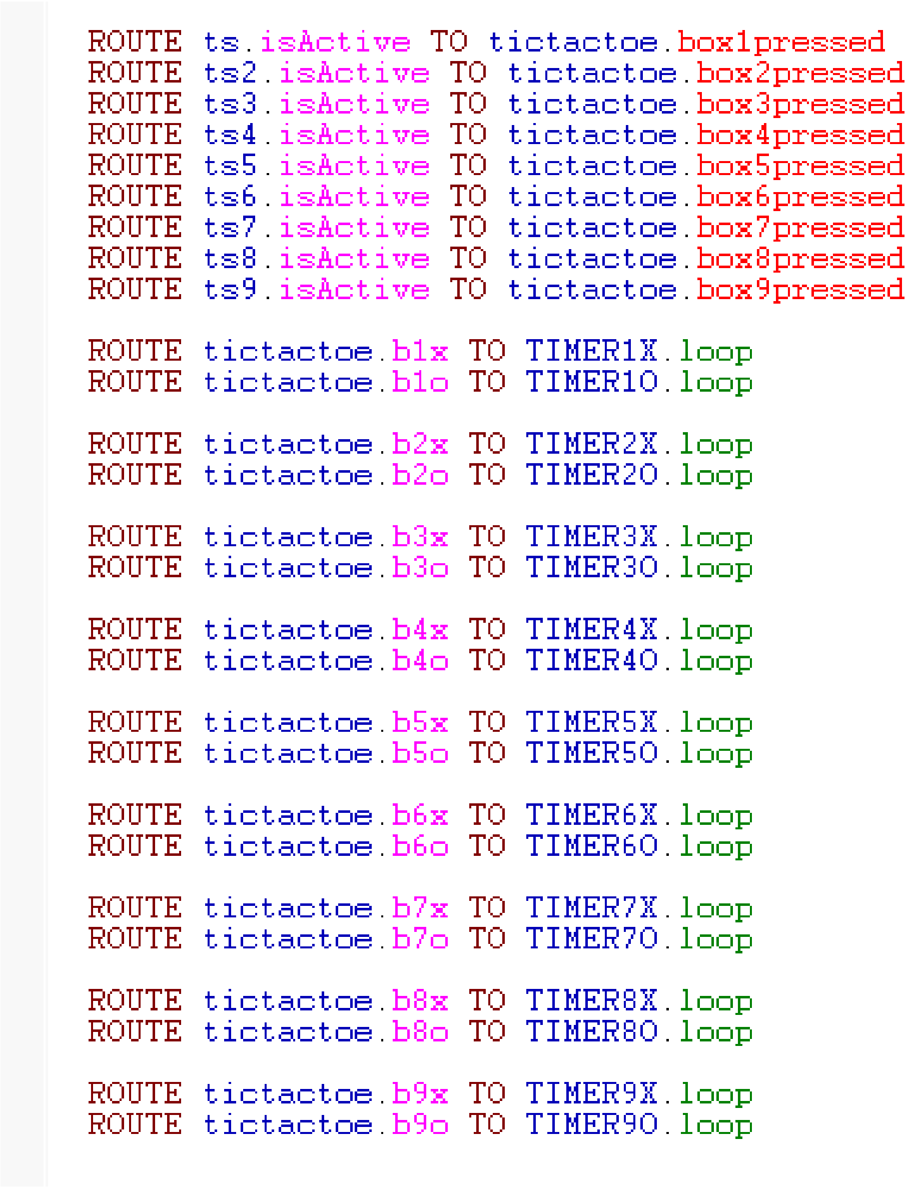


```
    function wins()
    {
       if ((b1=='x'&&b2=='x'&&b3=='x')||
           (b4=='x'&&b5=='x'&&b6=='x')||
           (b7=='x'&&b8=='x'&&b9=='x')||
           (b1=='x'&&b4=='x'&&b7=='x')||
           (b2=='x'&&b5=='x'&&b8=='x')||
           (b3=='x'&&b6=='x'&&b9=='x')||
           (b1=='x'&&b5=='x'&&b9=='x')||
           (b3=='x'&&b5=='x'&&b7=='x'))
       {
          sts.geometry.string[0]='X wins!';
          s.intensity=1;
       }

   else if((b1=='o'&&b2=='o'&&b3=='o')||
           (b4=='o'&&b5=='o'&&b6=='o')||
           (b7=='o'&&b8=='o'&&b9=='o')||
           (b1=='o'&&b4=='o'&&b7=='o')||
           (b2=='o'&&b5=='o'&&b8=='o')||
           (b3=='o'&&b6=='o'&&b9=='o')||
           (b1=='o'&&b5=='o'&&b9=='o')||
           (b3=='o'&&b5=='o'&&b7=='o'))
       {
           sts.geometry.string[0]='O wins!';
           s.intensity=1;

       }
 else if (count==9){
             sts.geometry.string[0]='Draw!';

       }
    }

    "]
}
```

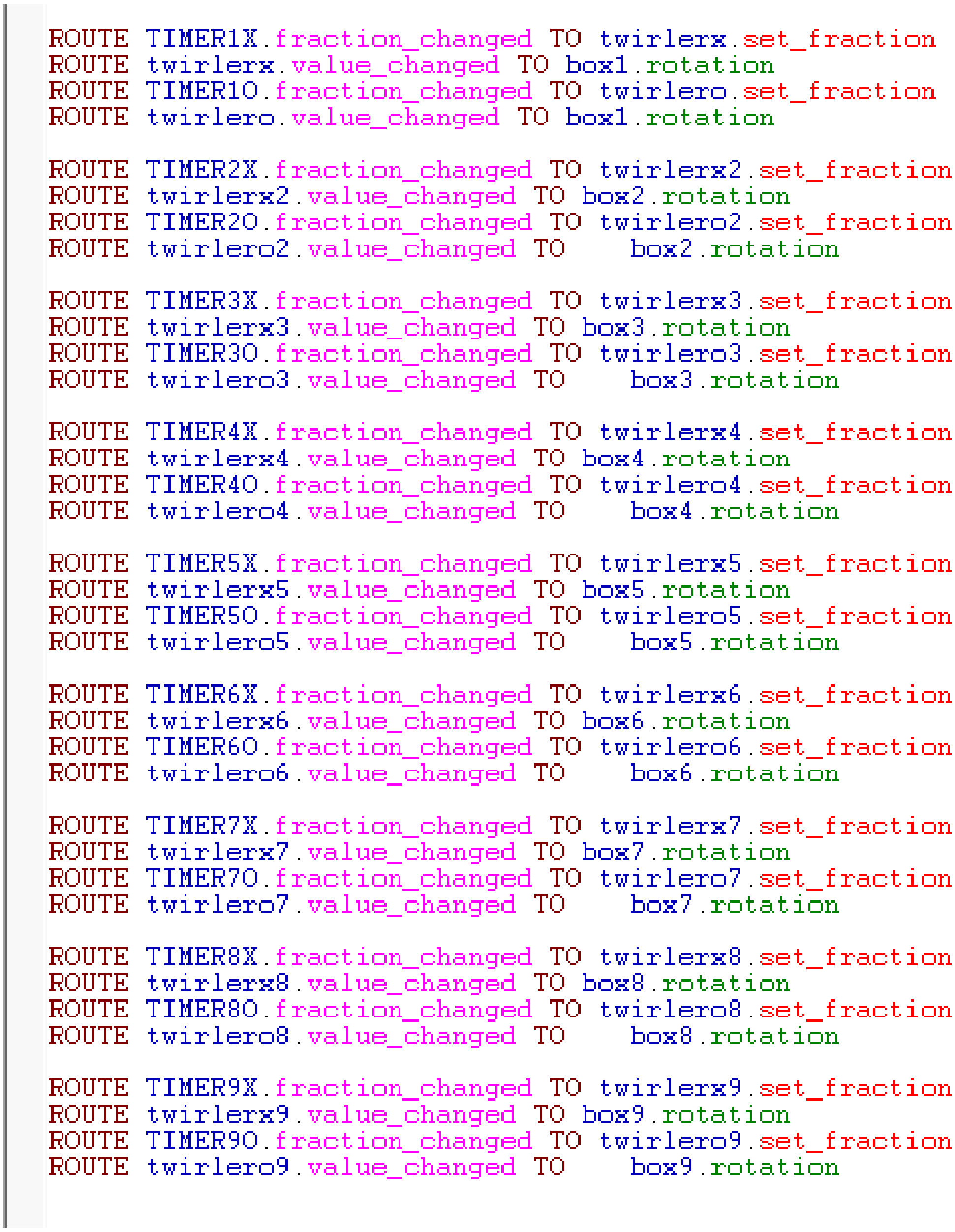
These are the ROUTE statements:



```
ROUTE sensor.isActive    TO tictactoe.initialize
ROUTE sensor.isActive   TO TIMERI.loop
ROUTE TIMERI.fraction_changed TO twirlerI.set_fraction
ROUTE twirlerI.value_changed TO box1.rotation
ROUTE twirlerI.value_changed TO box2.rotation
ROUTE twirlerI.value_changed TO box3.rotation
ROUTE twirlerI.value_changed TO box4.rotation
ROUTE twirlerI.value_changed TO box5.rotation
ROUTE twirlerI.value_changed TO box6.rotation
ROUTE twirlerI.value_changed TO box7.rotation
ROUTE twirlerI.value_changed TO box8.rotation
ROUTE twirlerI.value_changed TO box9.rotation
```



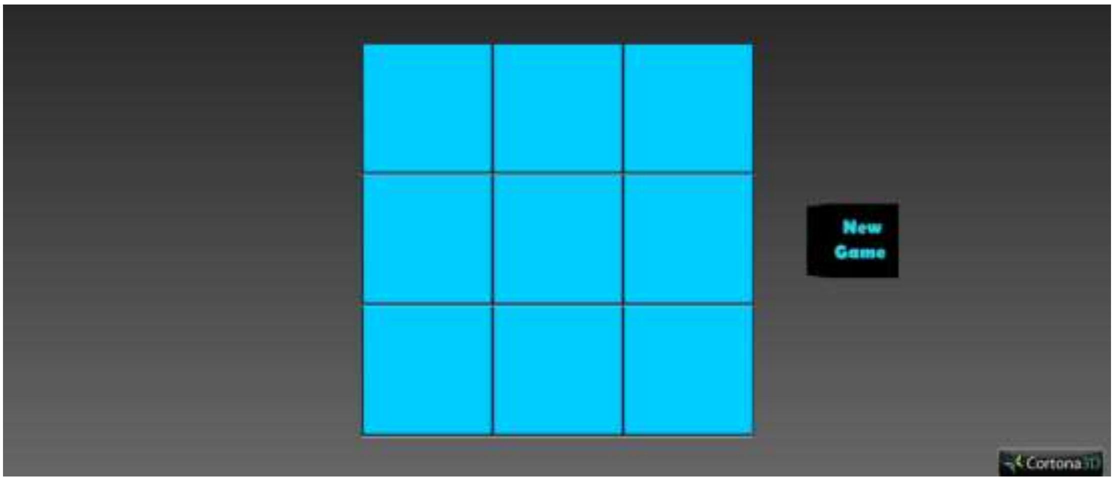
```
ROUTE ts.isActive TO tictactoe.box1pressed
ROUTE ts2.isActive TO tictactoe.box2pressed
ROUTE ts3.isActive TO tictactoe.box3pressed
ROUTE ts4.isActive TO tictactoe.box4pressed
ROUTE ts5.isActive TO tictactoe.box5pressed
ROUTE ts6.isActive TO tictactoe.box6pressed
ROUTE ts7.isActive TO tictactoe.box7pressed
ROUTE ts8.isActive TO tictactoe.box8pressed
ROUTE ts9.isActive TO tictactoe.box9pressed

ROUTE tictactoe.b1x TO TIMER1X.loop
ROUTE tictactoe.b1o TO TIMER1O.loop

ROUTE tictactoe.b2x TO TIMER2X.loop
ROUTE tictactoe.b2o TO TIMER2O.loop

ROUTE tictactoe.b3x TO TIMER3X.loop
ROUTE tictactoe.b3o TO TIMER3O.loop

ROUTE tictactoe.b4x TO TIMER4X.loop
ROUTE tictactoe.b4o TO TIMER4O.loop

ROUTE tictactoe.b5x TO TIMER5X.loop
ROUTE tictactoe.b5o TO TIMER5O.loop

ROUTE tictactoe.b6x TO TIMER6X.loop
ROUTE tictactoe.b6o TO TIMER6O.loop

ROUTE tictactoe.b7x TO TIMER7X.loop
ROUTE tictactoe.b7o TO TIMER7O.loop

ROUTE tictactoe.b8x TO TIMER8X.loop
ROUTE tictactoe.b8o TO TIMER8O.loop

ROUTE tictactoe.b9x TO TIMER9X.loop
ROUTE tictactoe.b9o TO TIMER9O.loop
```



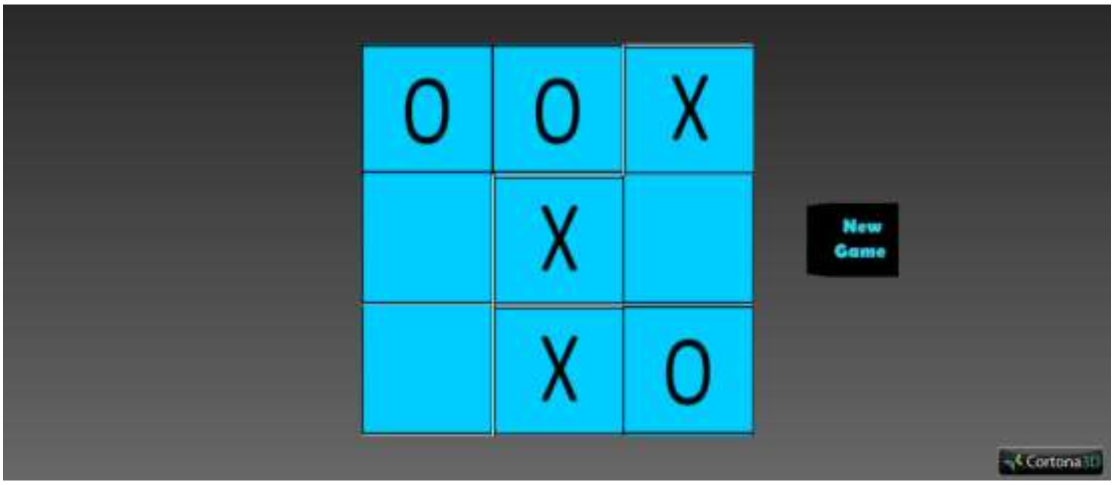
```
ROUTE TIMER1X.fraction_changed TO twirlerx.set_fraction
ROUTE twirlerx.value_changed TO box1.rotation
ROUTE TIMER1O.fraction_changed TO twirlero.set_fraction
ROUTE twirlero.value_changed TO box1.rotation

ROUTE TIMER2X.fraction_changed TO twirlerx2.set_fraction
ROUTE twirlerx2.value_changed TO box2.rotation
ROUTE TIMER2O.fraction_changed TO twirlero2.set_fraction
ROUTE twirlero2.value_changed TO    box2.rotation

ROUTE TIMER3X.fraction_changed TO twirlerx3.set_fraction
ROUTE twirlerx3.value_changed TO box3.rotation
ROUTE TIMER3O.fraction_changed TO twirlero3.set_fraction
ROUTE twirlero3.value_changed TO    box3.rotation

ROUTE TIMER4X.fraction_changed TO twirlerx4.set_fraction
ROUTE twirlerx4.value_changed TO box4.rotation
ROUTE TIMER4O.fraction_changed TO twirlero4.set_fraction
ROUTE twirlero4.value_changed TO    box4.rotation

ROUTE TIMER5X.fraction_changed TO twirlerx5.set_fraction
ROUTE twirlerx5.value_changed TO box5.rotation
ROUTE TIMER5O.fraction_changed TO twirlero5.set_fraction
ROUTE twirlero5.value_changed TO    box5.rotation

ROUTE TIMER6X.fraction_changed TO twirlerx6.set_fraction
ROUTE twirlerx6.value_changed TO box6.rotation
ROUTE TIMER6O.fraction_changed TO twirlero6.set_fraction
ROUTE twirlero6.value_changed TO    box6.rotation

ROUTE TIMER7X.fraction_changed TO twirlerx7.set_fraction
ROUTE twirlerx7.value_changed TO box7.rotation
ROUTE TIMER7O.fraction_changed TO twirlero7.set_fraction
ROUTE twirlero7.value_changed TO    box7.rotation

ROUTE TIMER8X.fraction_changed TO twirlerx8.set_fraction
ROUTE twirlerx8.value_changed TO box8.rotation
ROUTE TIMER8O.fraction_changed TO twirlero8.set_fraction
ROUTE twirlero8.value_changed TO    box8.rotation

ROUTE TIMER9X.fraction_changed TO twirlerx9.set_fraction
ROUTE twirlerx9.value_changed TO box9.rotation
ROUTE TIMER9O.fraction_changed TO twirlero9.set_fraction
ROUTE twirlero9.value_changed TO    box9.rotation
```

Game Screenshots:

A new game starts:

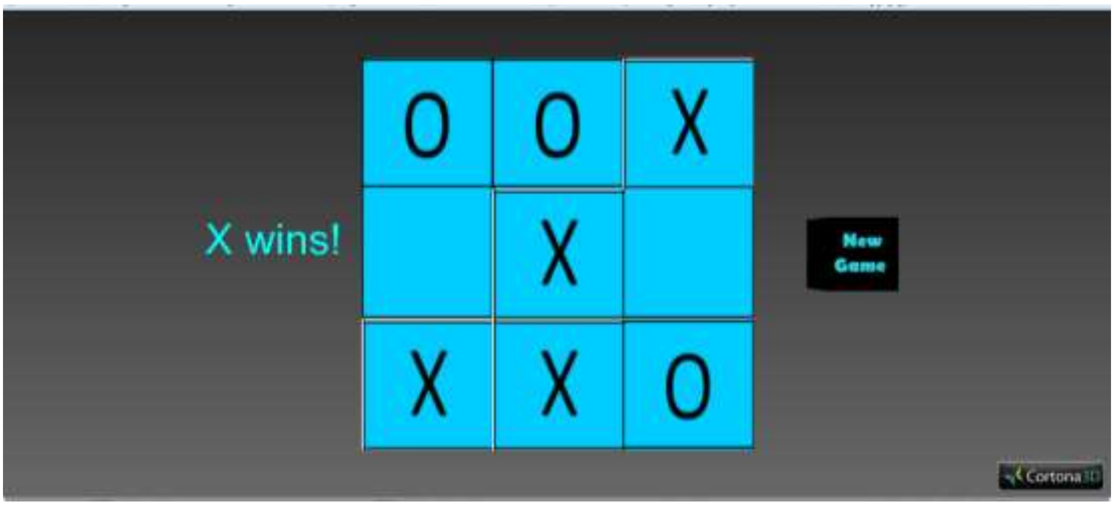


Screenshot at the middle of the game



O	O	X
	X	
	X	O

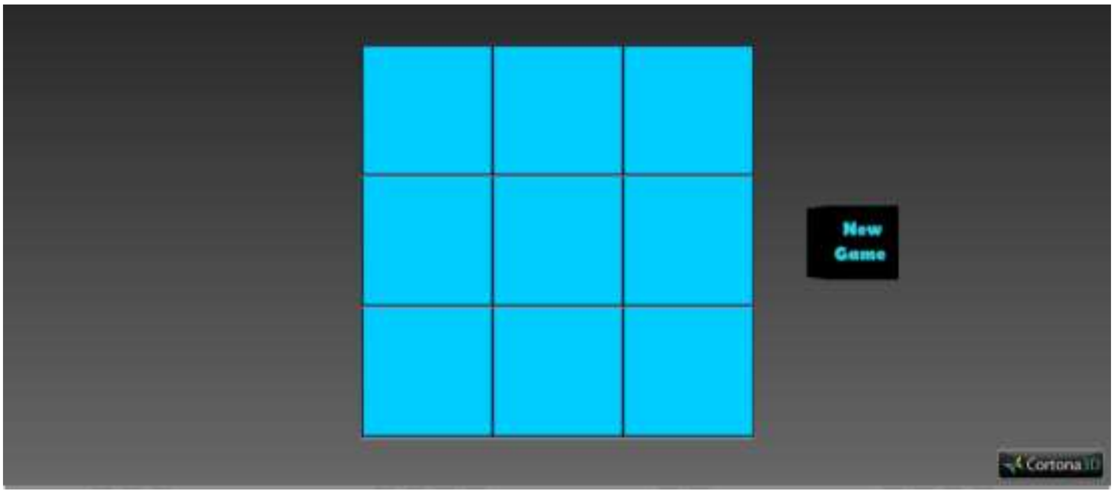
Player X plays and wins!



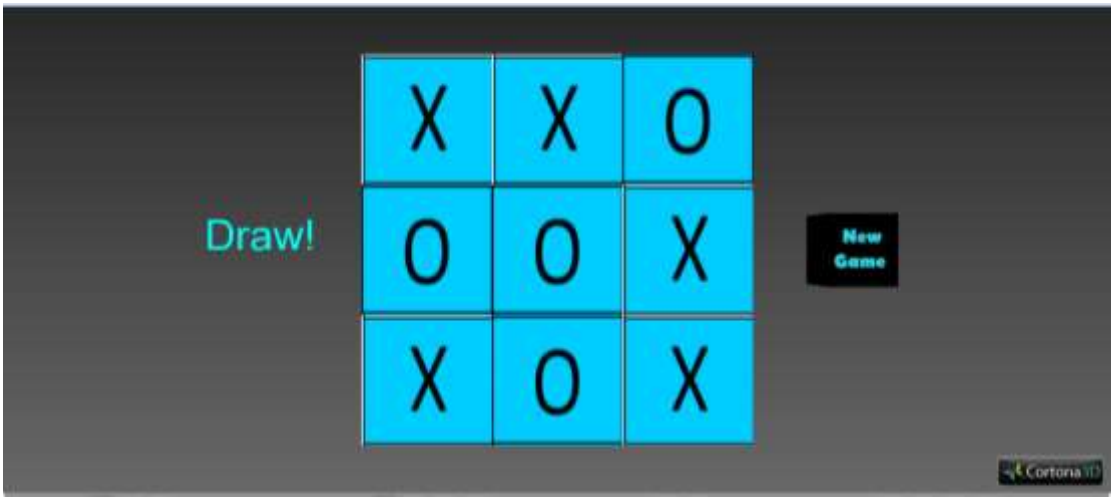
X wins!

O	O	X
	X	
X	X	O

The “New Game” button is pressed, and a new game starts



This time a draw occurs:



Draw!

X	X	O
O	O	X
X	O	X