```
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="content-type" content="charset=utf-8" />
      <link rel="stylesheet" type="text/css" href="styles/base.css" />
      <title>VisualizingJS - Robust - Fragile -mark breneman</title>
   </head>
   <body>
      <canvas id="myCanvas" width="600" height="600"></canvas>
      <script charset="utf-8" src="scripts/paper.js"></script>
      <script charset="utf-8" src="scripts/skip.js"></script>
      <script charset="utf-8" type="text/paperscript" canvas="myCanvas">

//SETTING UP THE SCENE:THIS IS NOT NECESSARY...JUST A STEWART CONVENTION
var background = new Path.Rectangle( view.center, view.bounds.width )
background.fillColor = 'black'
background.position = view.center
project.activeLayer.insertChild( 0, background )

//CREATE TIMING TEXT FOR SETUP
var durationSeconds = 60
var running = true

var timeText = new PointText(view.center);
   timeText.paragraphStyle.justification = 'center';
   timeText.characterStyle.fontSize = 20;
   timeText.fillColor = 'gray';
   timeText.position=view.size/2;

var frameCountText = timeText.clone();
   frameCountText.position.y+=30;
   frameCountText.fillColor = 'white';

var DiagnosticText = timeText.clone();
      DiagnosticText.position.y+=60;
      DiagnosticText.fillColor = 'red';


//CREATE CHARACTERS
var squareSide=60;
var offset = 10;
var squares = [];//An Array to Hold Characters
var rectStartPoint= new Point(165,offset+squareSide)//First Corner of First Square
var rectSize = new Size(squareSide,squareSide)//Set the size of the Squares

for(var i = 0; i<4; i++){
    var placementX=rectStartPoint.x+i*squareSide*(1+1/6)+offset;//Set the X-Position as a function of the for Loop
    var placementPoint= new Point(placementX,offset);//Bring in that X-Position
    var square = new Path.Rectangle(placementPoint,rectSize);
    square.fillColor = 'white';//Set the Fill Color
    square.destination = new Point(placementX+squareSide/2, view.size.height-squareSide/2-offset);
    squares.push(square);//Add the newly created squares to the array
 }
```

```
//SETTING UP A ROTATION PROPERTY FOR PATH OBJECTS SO THAT WE CAN KNOW WHERE OBJECTS ARE IN ROTATION
//CREATE A ROTATION ANGLE PROPERTY ASSOCIATED WITH PATHS
//CREATE A FUNCTION(CALLED LOGGED ROTATE) WHICH INCREASE THE ANGLE AND KEEPS TRACK BY INCREMENTING
//THE ROTATION ANGLE PROPERTY
 Path.prototype.rotationAngle = 0
 Path.prototype.loggedRotatePoint = function( changeInRotation, point ){

   this.rotationAngle += changeInRotation
   this.rotate( changeInRotation,point )
 }

 //SETTING UP A ROTATION PROPERTY FOR PATH OBJECTS SO THAT WE CAN KNOW WHERE OBJECTS ARE IN ROTATION
 //CREATE A ROTATION ANGLE PROPERTY ASSOCIATED WITH PATHS
 //CREATE A FUNCTION(CALLED LOGGED ROTATE) WHICH INCREASE THE ANGLE AND KEEPS TRACK BY INCREMENTING
 //THE ROTATION ANGLE PROPERTY
  Path.prototype.rotationAngle = 0
  Path.prototype.loggedRotate = function( changeInRotation, point ){

   this.rotationAngle += changeInRotation
   this.rotate( changeInRotation)
  }

var gravity=9.8;
var damping=.4;
var bounced=false;

var vectorVelocity= new Point(0,0);

var onFrame = function(event){

      //KEEPING TRACK OF TIME
       if( event.time > durationSeconds && running === true ){

        running = false
        squares[0].remove()
        squares[1].remove()
        squares[2].remove()
        squares[3].remove()
        background.remove()
        // document.title = 'Paper finished.'
       }

       // else if( running === true ) {
       //
       //         document.title = 'Animation at '+ event.time.floor().toPaddedString( 2 ) +' seconds.'
       //        }
       // timeText.content = Math.round(event.time) + " seconds";
       //        frameCountText.content = Math.round(event.count) + " frames";

       // DiagnosticText.content = squares[0].position;

       //ENTRANCE
```

```
if(event.time.isBetween(0,2)){

  for(i=0; i<squares.length; i++){

   // var vectorAcceleration;
     vectorVelocity.y += gravity;
     vectorVelocity.y *= damping;

     squares[i].position += vectorVelocity;
     if( squares[i].position.y>view.size.height-squareSide/2||squares[i].position.y<squareSide/2){

         vectorVelocity.y *= .7;
         damping*=-.9;
         }}
  //This doesnt' work as planned but I'll take it...

  for(i=0; i<squares.length; i++) {
    var vector =  squares[i].destination - squares[i].position

    squares[i].position += vector/100//This is an easing
    if( vector.length < 2 ){
    squares[i].position = squares[i].destination;
                }}
}

//SPREAD OUT
if(event.time.isBetween(2,7)){
for(i=0; i<=1; i++) {
squares[i].destination.x = squareSide/2+offset+i*(offset+squareSide)
var vector =  squares[i].destination - squares[i].position
squares[i].position+=vector/70;
}

for(j=3; j>=2; j--) {
squares[j].destination.x = view.size.width-squareSide*4-offset+j*(offset+squareSide)
var vector =  squares[j].destination - squares[j].position
squares[j].position+=vector/70;
}
}

//ROTATE TO STACK
if(event.time.isBetween(7,10)){
var point = squares[0].bounds.topRight;
var pointOffset = new Point(point.x+5,point.y-10)
squares[1].loggedRotatePoint(-1,pointOffset)
squares[1].rotationAngle=0

var point2 = squares[3].bounds.topLeft;
var point2Offset = new Point(point2.x-5,point2.y-10)
squares[2].loggedRotatePoint(1,point2Offset)
squares[2].rotationAngle=0
}
```

```
 if(event.time.isBetween(11,11.125)){
 squares[1].position.y+=1
 squares[2].position.y+=1
 }

 //BACK TO CENTER
 if(event.time.isBetween(11.5,15.2)){
 squares[0].position.x+=1
 squares[1].position.x+=1

 squares[2].position.x-=1
 squares[3].position.x-=1
 }

 //ROTATE TO STACK RIGHT
 if(event.time.isBetween(16,19)){
 var point = squares[2].bounds.topLeft;
 var pointOffset = new Point(point.x-5.5,point.y-10)
 squares[1].loggedRotatePoint(1,pointOffset)
 squares[1].rotationAngle=0
 }
 if(event.time.isBetween(19.5,19.65)){
 squares[1].position.y+=1
 }

 if(event.time.isBetween(20,23)){
 var point = squares[2].bounds.topLeft;
 var pointOffset = new Point(point.x-5.5,point.y-10)
 squares[1].loggedRotatePoint(-1,pointOffset)
 squares[1].rotationAngle=0
 }

 if(event.time.isBetween(24,24.125)){
 squares[1].position.y+=1
 }

 //ROTATE TO STACK LEFT
 if(event.time.isBetween(25,28)){
 var point = squares[1].bounds.topRight;
 var pointOffset = new Point(point.x+5.5,point.y-10)
 squares[2].loggedRotatePoint(-1,pointOffset)
 squares[2].rotationAngle=0
 }
 if(event.time.isBetween(28.25,28.375)){
 squares[2].position.y+=1
 }

if(event.time.isBetween(29,32)){
var point = squares[1].bounds.topRight;
var pointOffset = new Point(point.x+6,point.y-10)
squares[2].loggedRotatePoint(1,pointOffset)
squares[2].rotationAngle=0
}
```

```
 if(event.time.isBetween(33,33.125)){
  squares[2].position.y+=1
}

//MOVE AND ROTATE TO ROBUST!
if(event.time.isBetween(34,35.35)){
squares[1].position.x+=.5
squares[2].position.x+=.5
}

 if(event.time.isBetween(35.30,38.31)){
 var point = squares[1].bounds.topRight;
 var pointOffset = new Point(point.x+6,point.y-10)
 squares[2].loggedRotatePoint(-1,pointOffset)
 squares[2].rotationAngle=0
 }
 if(event.time.isBetween(39,39.125)){
 squares[2].position.y+=1
 }

 //SETUP FRAGILE
 if(event.time.isBetween(42,45)){
 var point = squares[1].bounds.topRight;
 var pointOffset = new Point(point.x+6,point.y-10)
 squares[2].loggedRotatePoint(+1,pointOffset)
 squares[2].rotationAngle=0
 }
 if(event.time.isBetween(46,46.125)){
 squares[2].position.y+=1
 }
 if(event.time.isBetween(47,48.35)){
     squares[1].position.x-=.5
     squares[2].position.x-=.5
     }
 if(event.time.isBetween(48.5, 51.5)){
   var group = new Group([squares[2], squares[3]]);
   var point = squares[1].bounds.topRight;
   var pointOffset = new Point(point.x+6,point.y-10)
   group.rotate(-1,pointOffset)
 }

 if(event.time.isBetween(51.5, 51.75)){
   squares[0].loggedRotate(Math.floor((Math.random()*1)+1))
   squares[1].loggedRotate(Math.floor((Math.random()*-2)+1))
   squares[2].loggedRotate(Math.floor((Math.random()*3)+1))
   squares[3].loggedRotate(Math.floor((Math.random()*-4)+1))
 }

  if(event.time.isBetween(52, 55) && squares[0].position.x<view.size.width/2){
     squares[0].position.x+=.125
     squares[1].position.x+=.125
     squares[2].position.x+=.125
```

```
        squares[3].position.x+=.125
    }
    if(event.time.isBetween(58, 60) && squares[0].position.x<view.size.width/2){
        squares[0].fillColor.brightness -= .01
        squares[1].fillColor.brightness -= .01
        squares[2].fillColor.brightness -= .01
        squares[3].fillColor.brightness -= .01
    }

};
    </script>
  </body>
</html>
```