# Uzdevuma koda karkas.

<!DOCTYPE html>  
<html class="no-js compliant lang\_eng" lang="eng">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">  
<meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">  
<style>  
  
</style>  
</head>  
<body>  
  
<canvas id="debug" style="position:absolute; left:10px"></canvas>  
<script src="js/Three.js" type="text/javascript" ></script>  
  
<script>

</script>

# Inicializējam visus mainīgos

var container;  
 var camera, scene, renderer;  
 var plane;  
 var targetRotation = 0;  
 var windowHalfX = window.innerWidth / 1;  
 var windowHalfY = window.innerHeight / 1;  
 var particleImage = new Image();  
  
 init();  
 animate();

# Init funkcija

Izveidojam div conteineri , kurā glabāsies karuselis. Kā arī uzstādām kur kamera stāvēs un izveidojam three.js scēnu un pievienojam.

container = document.createElement( 'div' );  
 document.body.appendChild( container );  
  
 camera = new THREE.PerspectiveCamera( 60, window.innerWidth / window.innerHeight, 30, 2500 ); // FlyCamera // FlyControls  
  
 camera.position.y = 60;  
 camera.position.z = 500;  
 scene = new THREE.Scene();  
 parent = new THREE.Object3D();  
 parent.position.y = 60;  
 scene.add( parent );

Izveidojam “masīvu” priekš bildēm –

var arImgRotator = {  
 "0":"1.jpg",  
 "1":"2.jpg",  
 "2":"3.jpg",  
 "3":"4.jpg",  
 "4":"5.jpg",  
 "5":"1.jpg",  
 "6":"7.jpg",  
 "7":"8.jpg",  
 "8":"9.jpg",  
 "9":"10.jpg",  
 "10":"11.jpg",  
 "11":"12.jpg"  
 }

Norādam cik kopā bildes būs, kāds izmērs, ap kuru asi rotēs, kāda pozīcijā sāks ( grādos ).

var camSize = 100; // lielums   
 var startAngle = 0; // kada pozicija bildes sak radit  
 var circleRadius = 230; //radius pasam karuselim  
  
 var centerX = -5; // ap kuru asi rotēs x / z un kura vieta atordas abi punkti  
 var centerZ = -0.5  
  
 var mpi = Math.PI/180;  
 var startRadians = startAngle;  
 var totalSpheres = 11;  
 var incrementAngle = 360/totalSpheres;  
 var incrementRadians = incrementAngle \* mpi;

Funkcija kas apstrādā bildes un planes. Pievieno tās.

for ( var i = 0; i < totalSpheres; i ++ ) {  
  
 var xp = centerX + Math.sin(startRadians) \* circleRadius;  
 var zp = centerZ + Math.cos(startRadians) \* circleRadius;  
  
 var planObj = new THREE.Mesh(  
 new THREE.PlaneGeometry( camSize, 1\*camSize), // plane izmers  
 new THREE.MeshBasicMaterial( {  
 map: THREE.ImageUtils.loadTexture( 'img/'+arImgRotator[i]),  
 doubleSided: false,  
 wireframe: false,  
 overdraw: true  
 })  
  
  
 );  
  
 planObj.position.x = xp;  
 planObj.position.z = zp;  
 planObj.rotation.y = i\*incrementAngle \* (Math.PI/180);  
 startRadians += incrementRadians;  
  
 parent.add( planObj );  
  
  
 }

izveidojam canvas renderē objektu un pievienoja peles funkcijas.

// Render canvas  
   
 renderer = new THREE.CanvasRenderer( { /\*antialias: true\*/ } ); // WebGLRenderer  
 renderer.setSize( window.innerWidth, window.innerHeight );  
 renderer.setClearColorHex( 0xfffFFF, 100 );  
 container.appendChild( renderer.domElement );  
  
 //peles listeneri  
 document.addEventListener( 'mousedown', onDocumentMouseDown, true );  
 document.addEventListener( 'touchstart', onDocumentTouchStart, true );  
 document.addEventListener( 'touchmove', onDocumentTouchMove, true );

Funkcija, kas nolasa listener no mouse pogām. Un, uzstāda ātrumu cik ātri bildes kustēsies ap asi.

function onDocumentMouseMove( event ) {

mouseX = event.clientX - windowHalfX;

mouseY = event.clientY - windowHalfY;

targetRotation = targetRotationOnMouseDown + ( mouseX - mouseXOnMouseDown ) \* 0.01; //atrums cik atri bildes iet ap asi

render();

}

Funkcija, ar kuras palidzību nav animācija ja listeners nav nospiests

function onDocumentMouseUp( event ) {

document.removeEventListener( 'mousemove', onDocumentMouseMove, false );

document.removeEventListener( 'mouseup', onDocumentMouseUp, false );

document.removeEventListener( 'mouseout', onDocumentMouseOut, false );

}

function onDocumentMouseOut( event ) {

document.removeEventListener( 'mousemove', onDocumentMouseMove, true );

document.removeEventListener( 'mouseup', onDocumentMouseUp, true );

document.removeEventListener( 'mouseout', onDocumentMouseOut, true );

}

function onDocumentTouchStart( event ) {

if ( event.touches.length == 1 ) {

event.preventDefault();

mouseXOnMouseDown = event.touches[ 0 ].pageX - windowHalfX;

targetRotationOnMouseDown = targetRotation;

}

}

Renderējam Scenu un izsaucam animācijas loop

function animate() {

requestAnimationFrame( animate );

render();

}

function render() {

parent.rotation.y += ( targetRotation - parent.rotation.y ) \* 0.05;

renderer.render( scene, camera );

}