BRICK BREAKER

DESTROY ALL THE BLOCKS

THREE.JS

Open source library for webGL easy to use

Creating a cube in webGL takes more than one hundred lines of code in Javascript and GLSL, while in Three.js requires few lines

BLOCKS

The main elements of the game are bricks

createLine(depth,emptyIndex,opposite,doubleHit)

createBlock(x,y,col)

block = new THREE.Mesh(new THREE.CubeGeometry(Width,Height,Depth, Quality,Quality,Quality),Material);

LAMBERT

var blockMaterial = new THREE.MeshLambertMaterial({color: col});

MATERIAL PROPERTIES:

#.color

Diffuse color of the material. Default is white.

#.ambient

Ambient color of the material, multiplied by the color of the AmbientLight. Default is white.

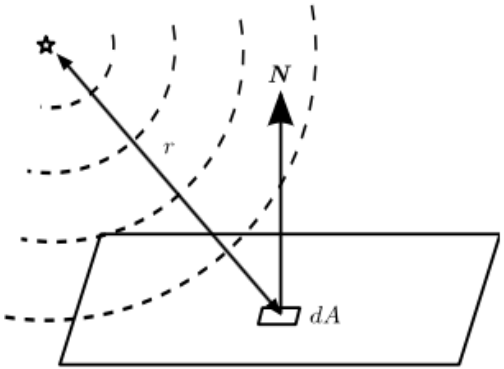
#.emissive

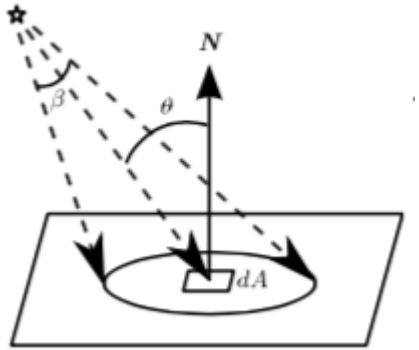
Emissive (light) color of the material, essentially a solid color unaffected by other lighting. Default is black.

LIGHT

var pointLight = new THREE.PointLight(0xFFFFD0);

var spotLight = new THREE.SpotLight(0xFFFFD0);





TEXTURE

var planeMaterial = new THREE.MeshLambertMaterial({ map: THREE.ImageUtils.loadTexture('IMG/table.jpg') });

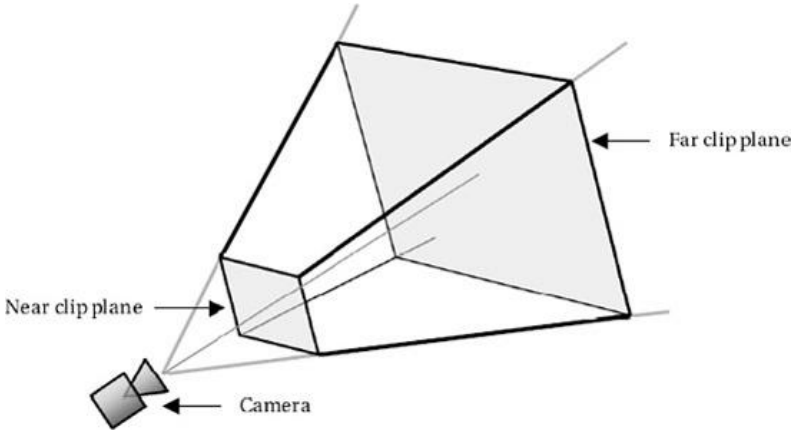
MATERIAL PROPERTIES:

#.map

Set color texture map. Default is null.

CAMERA

var camera = new THREE.PerspectiveCamera(VIEW_ANGLE, ASPECT, NEAR, FAR);



SHADOW

renderer.shadowMapEnabled = true;
renderer.shadowMapType = THREE.PCFSoftShadowMap;

block.receiveShadow = true;
block.castShadow = true;

spotLight.castShadow = true;

DRAW

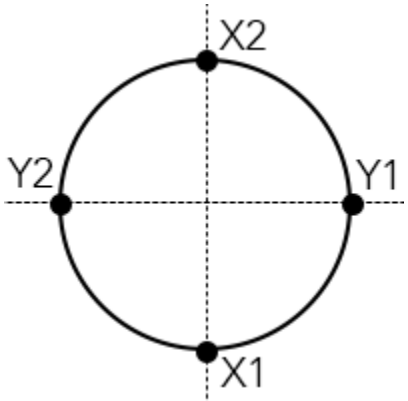
```
function createScene(){
...
renderer = new THREE.WebGLRenderer();
…
}
function draw(){
        renderer.render(scene, camera);
        requestAnimationFrame(draw);
        ...
}
```

BALL

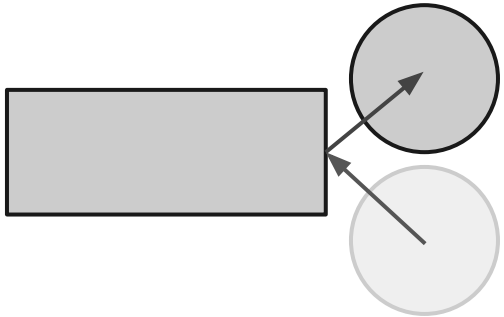
*var limitX = (initialWidth+5)-(10*dimensionMatrixBlock);*

function ballPhysics(){ … }

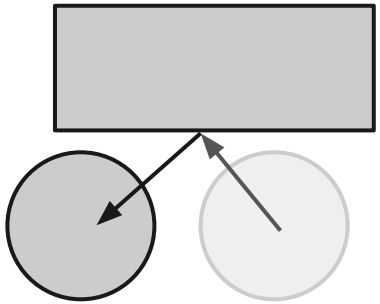
function checkPositionBall(positionX, positionY){ … }



BOUNCE



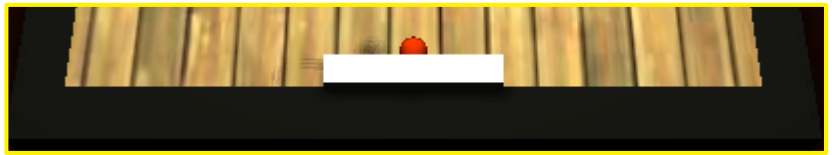
Lateral bounce → modifies only ballDirY



Frontal bounce → modifies only ballDirX

PADDLE

```
function paddlePhysics(){
    if ( (ball.position.x <= paddlePlayer.position.x + paddleWidth) && (ball.position.x >= paddlePlayer.position.x)){
        if ((ball.position.y <= paddlePlayer.position.y + paddleHeight/2) && (ball.position.y >= paddlePlayer.position.y - paddleHeight/2)){
            if (ballDirX < 0){
                paddlePlayer.scale.y = 1.5;
                ballDirX = -ballDirX;
                ballDirY -= paddlePlayerDirY;
            }
}
```



KEYBOARD

window.addEventListener('keyup', function(event) { Key.onKeyup(event); }, false);
window.addEventListener('keydown', function(event) { Key.onKeydown(event); }, false);

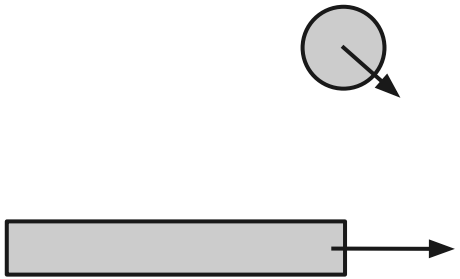
A = 65 | D = 68 | SPACE = 32 | W = 87 | S = 83

A - move left D - move right

SPACE - go W = next level S - start / reset

AUTOMATIC GAME

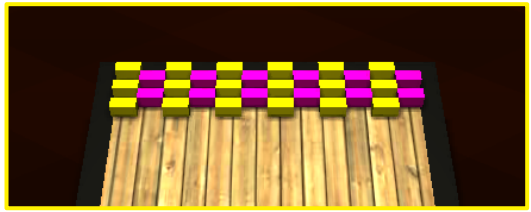
```
function automaticMovement(){
        paddlePlayerDirY = (ball.position.y - paddlePlayer.position.y) ;
        ...
}
```



LEVELS

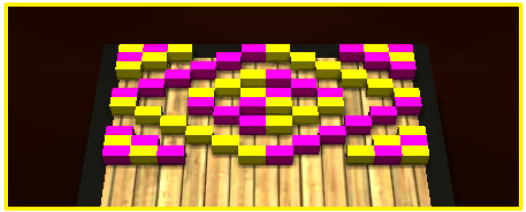
LEVEL 1

LEVEL 2

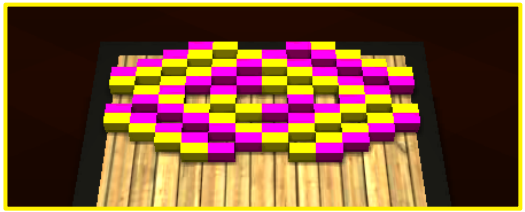


LEVEL 3

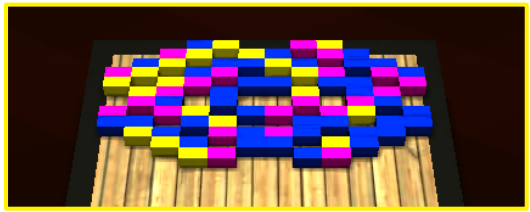
LEVEL 4



LEVEL 5



LEVEL 6



ONE HIT

DOUBLE HIT

BIBLIOGRAPHY

- Three.js http://threejs.org/docs/index.html#Manual/Introduction/Creating_a_scene
- CubeGeometry http://threejs.org/docs/#Reference/Extras.Geometries/BoxGeometry
- MeshLambertMaterial http://threejs.org/docs/#Reference/Materials/MeshLambertMaterial
- PointLight & SpotLight
  "Introduction to Computer Graphics A practical Learning Approach"
- Material http://threejs.org/docs/#Reference/Materials/Material
- Shadow http://learningthreejs.com/blog/2012/01/20/casting-shadows/
- Render http://threejs.org/docs/#Reference/Renderers/WebGLRenderer

POWERED BY

Silvia Fortunato

Nazzareno Marziale

Francesco Nobilia