UVOD U RACUNARSKU GRAFIKU

NA WEB-U

VECERAS PROLAZIMO

JAVASCRIPT

CANVAS

2D GRAFIKA

3D GRAFIKA

C--> JS

INT MAIN()

```c
#include <stdio.h>
int main() {
    return 0;
}
```

==

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script src="/putanja/do/fajla/main.js"></script>
    <title>Demo</title>
</head>
<body>

</body>
</html>
```

PROMENLJIVE,PRINTF, SCANF

```
// dinamcki jezik
// ne naglasavamo tipove
var broj = 2;
var ime  = "pera";
var drugoIme = prompt( 'Unesi ime...' ); // scanf
console.log(ime); // printf
console.log(drugoIme);
```

IF THEN ELSE

```
if(sati < 12 ){
    console.log('dobro jutro!');
} else if(sati > 12 && sati < 18) {
    console.log('dobar dan!');
} else {
    console.log('dobro vece');
}
```

PETLJE

```
for(var i =0; i < arr.length; i++){
    console.log("Zdravo " + arr[i]);
}
```

```
unos = prompt( 'Pogodi komandu za izlaz' );
while(unos !== 'exit') {
    console.log('idemo opet..' );
    unos = prompt( 'Opet probaj komandu' );
}
alert('bravo'); // printf v2
```

FUNKCIJE

```
function obavesti(poruka) {
    alert(poruka);
}
```

```
//anonimne funkcije
var obavesti = function (poruka) {
    alert(poruka);
}
```

```
function primeni(f, args) {
    f(args);
}
```

OOP U JS

```
// objekti se mogu praviti funkcijama
function Student(ime, prezime, index ) {
    this.ime = ime;
    this.prezime = prezime;
    this.index = index;
    this.getIme = function (){
        return this.ime;
    }
}

var pera = new Student('pera', 'peric', 'mn12345');
console.log(pera.getIme());
```

OOP U JS

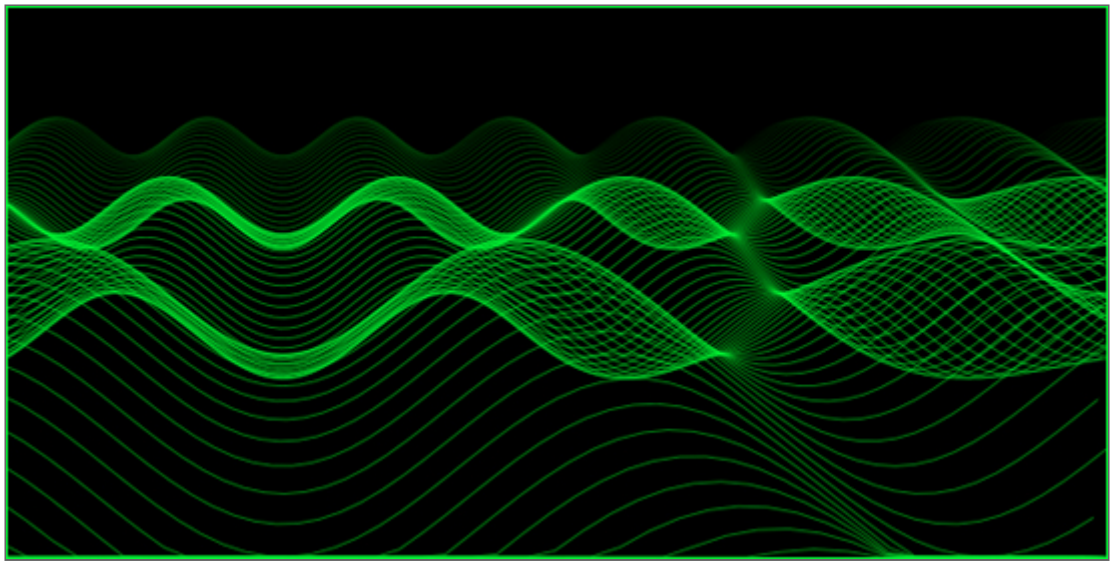
```
// JSON == JavaScript Object Notation
var pera = {
    ime: 'pera',
    prezime: 'peric',
    pozdrav: function (){
        alert("Pozdrav od Pere.");
    }
};
console.log(pera[ime]);
pera.pozdrav();
```

JS + HTML

```
// selektovanje elemenata
var text = document.getElementById('text');
text.style.color = "green"; // menjamo element
console.log(text.inner); // citamo iz elementa
```

CANVAS

KORISTI ZA GRAFIKU



CANVAS

POSTAVKA CANVAS ELEMENTA

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script src="js/main.js"></script>
    <title>Demo</title>
</head>
<body>
    <canvas id="canvas"></canvas>
    <!--  BITNO SKRIPTA SE UCITVA TEK POSLE DEFINISANJA CANVAS-A
    -->
    <script src="js/grafika.js"></script>
</body>
</html>
```

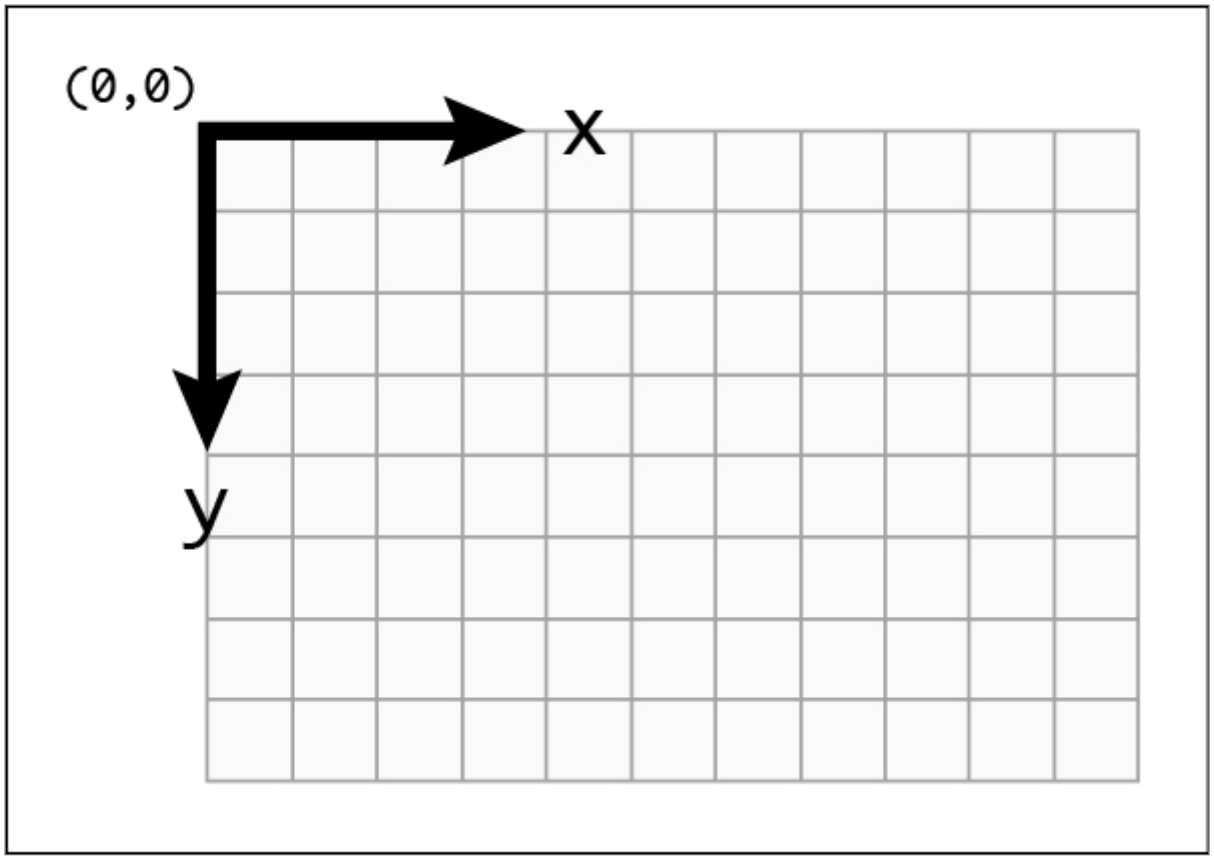
JS + CANVAS

```
// grafika.js
var canvas = document.getElementById( 'canvas');
var ctx = canvas.getContext( '2d'); // moze i 3D
canvas.height = 500;
canvas.width  = 500;
```

PARALELOGRAM

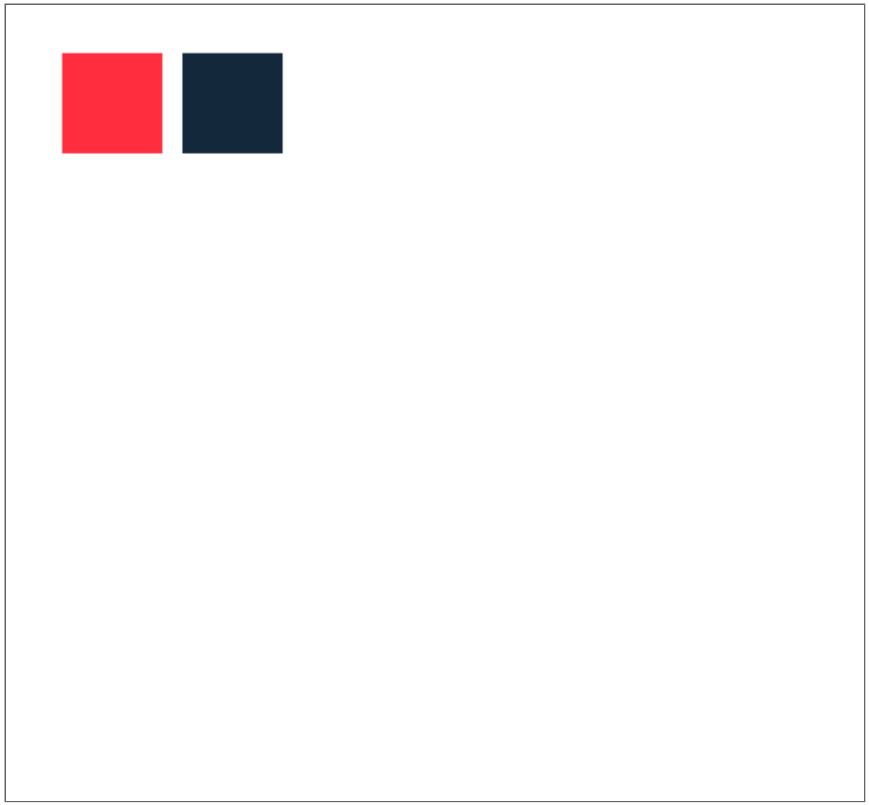
```
// vec imamo ctx
ctx.fillRect(20,20,100,200);
```

KOORDINATE



BOJENJE

```
// vec imamo kontekst u ctx
ctx.fillStyle = '#ff2d3d';
ctx.fillRect(20, 20, 50, 50);
ctx.fillStyle = rgb(20, 40, 60);
ctx.fillRect(80, 20, 50, 50);
```



PIRAMIDA

```
for(var i = 0; i < 15; i++)
    for(var j = 0; j < i; j++){
        if(i % 2 === 0)
            ctx.fillStyle =  '#ff2d3d';
        else
            ctx.fillStyle =  'rgb(20, 40, 60)';
        ctx.fillRect(j*25, i*25, 20,20);
    }
```

ANIMACIJA

```
var x = 0, y = 20;
var xspeed = 2, yspeed = 2;
function draw(){
    ctx.fillRect(x,y,  10, 10);
    x += xspeed;
    y += yspeed;
}
setInterval(draw,  30);
```


gif animacije

BOLJA ANIMACIJA

```
var x = 0, y = 20;
var xspeed = 2, yspeed = 2;
function draw(){
    ctx.fillStyle = "#232d3d";
    ctx.fillRect( 0, 0, canvas.width, canvas.height);
    ctx.fillStyle = "red";
    ctx.fillRect( 0, 0, canvas.width, canvas.height);
    ctx.fillRect(x,y,  10, 10);
    x += xspeed;
    y += yspeed;
}
setInterval(draw,  30);
```


gif animacije

P5.JS

OSNOVE

```
function setup(){
    // izvrsava se jednom na pocetku
}

function draw() {
    // izvarsava se 60 puta u sekundi
}
```

OSNOVE

```
function setup(){
    createCanvas(640, 480);
}

function draw() {
    ellipse(32, 32, 25, 50);
}
```

TRANSFORMACIJE

```
function setup(){
    createCanvas(640, 480);
}

function draw() {
    transform(width/2, height/2);
    scale(2);
    rotate(30);
    ellipse(64, 64, 20, 20);
}
```

OBJEKTI

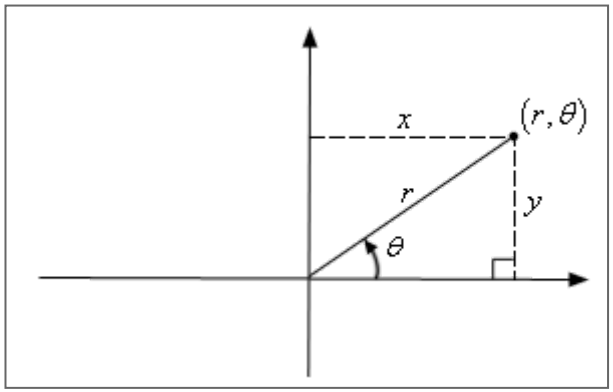
```
ellipse(x, y, width, height);
rect(x, y, width, height);
line(x1, y1, x2, y2);
triangle(x1, y1, x2, y2, x3, y3);
quad(x1, y1, x2, y2, x3, y3, x4, y4);
```

PARAMETARSKE JEDNACINE

```
x  = g(t)
y  = h(t)
```

```
function x(t) {
    return sin(t);
}
function y(t) {
    return cos(t);
}
```

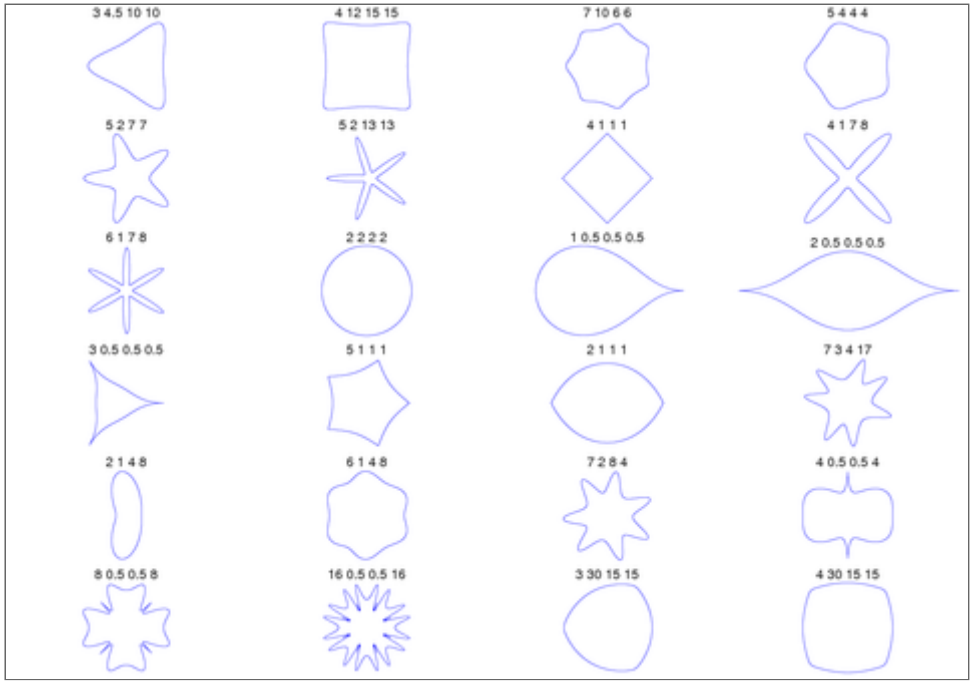
POLARNE KOORDINATE



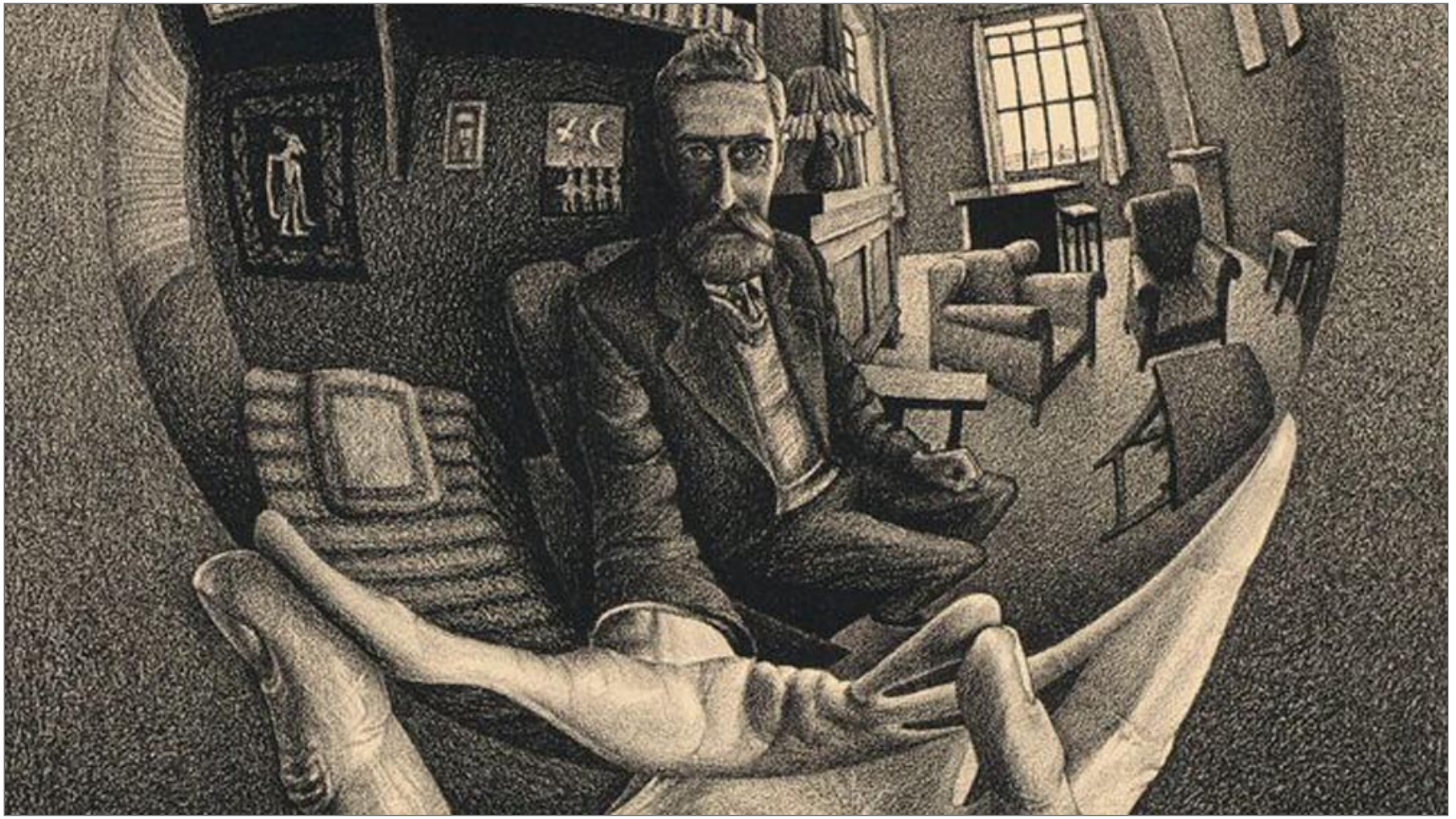
SUPERFORMULA

$$y = \sin(\phi) \cdot \left(\left| \frac{\cos\left(\frac{m \cdot \phi}{4}\right)}{a} \right|^{n_2} + \left| \frac{\sin\left(\frac{m \cdot \phi}{4}\right)}{b} \right|^{n_3} \right)^{-\left(\frac{1}{n_1}\right)}$$

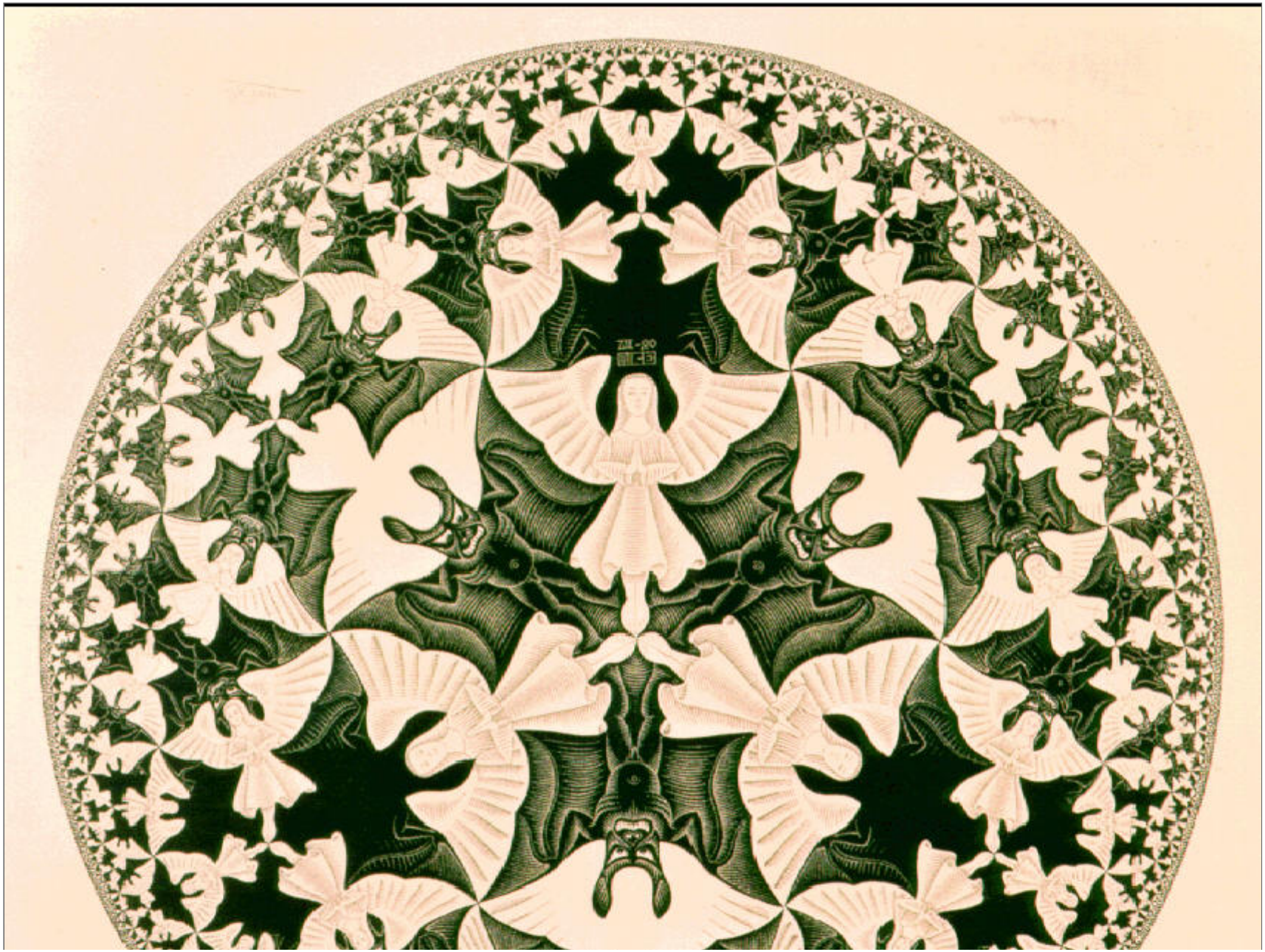
SUPERFORMULA

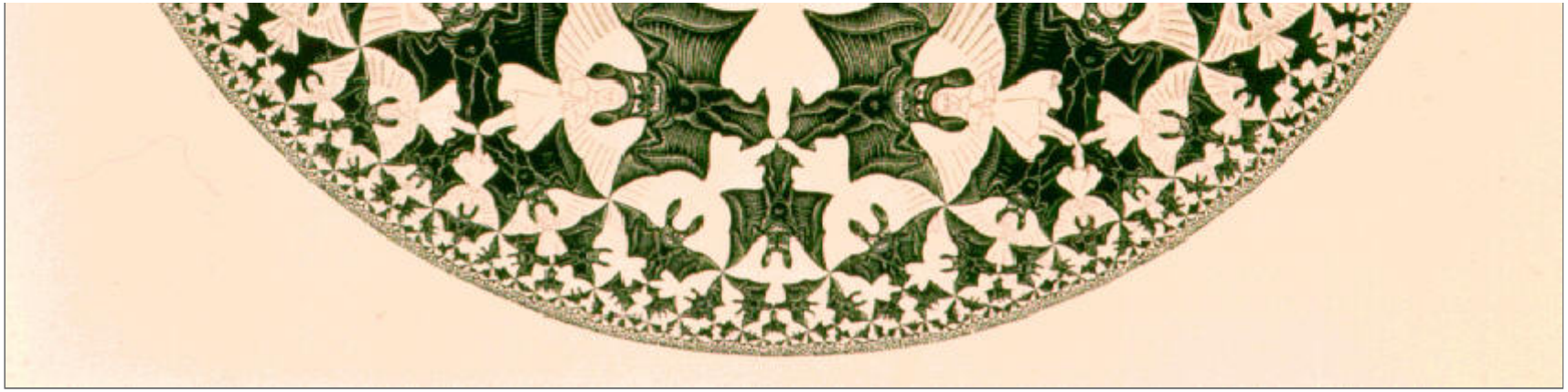


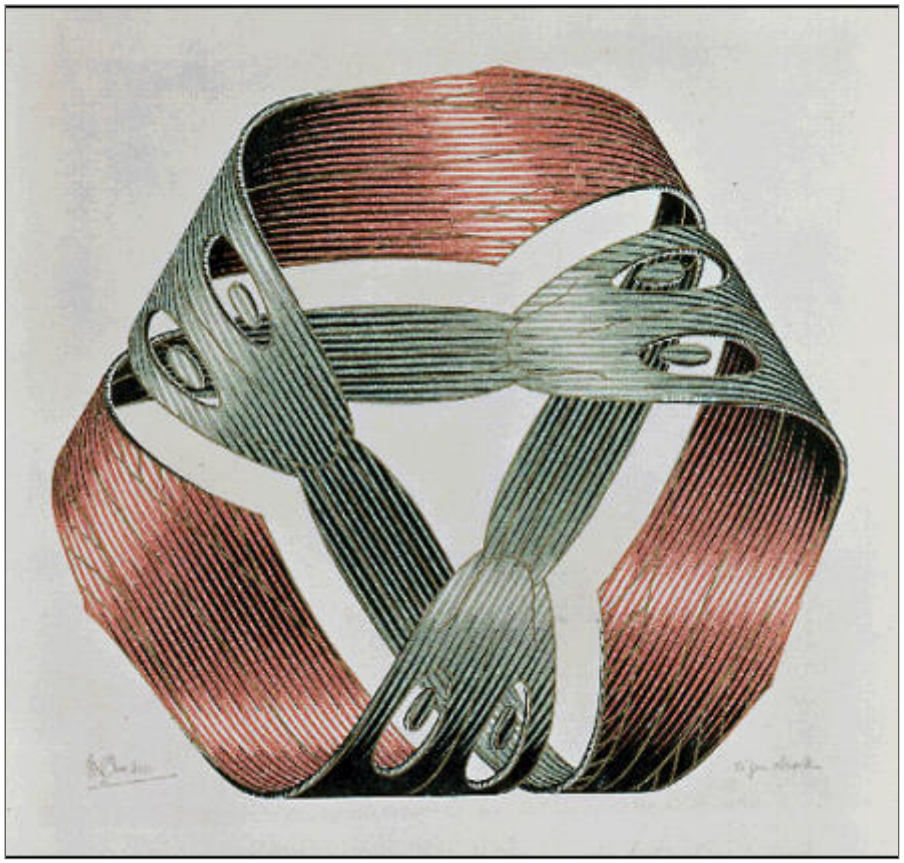
M.C ESCHER



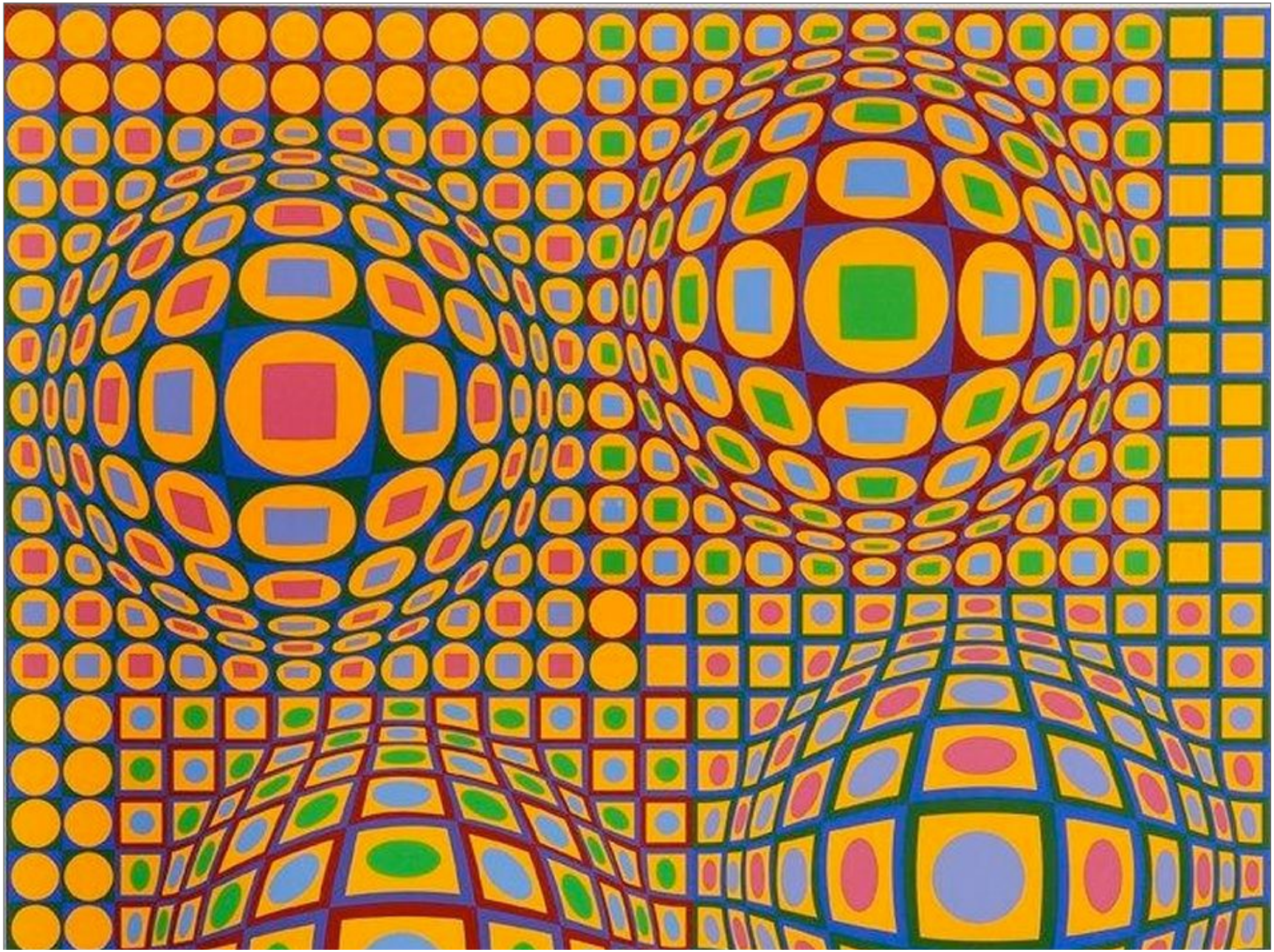

MCE
VII-'53

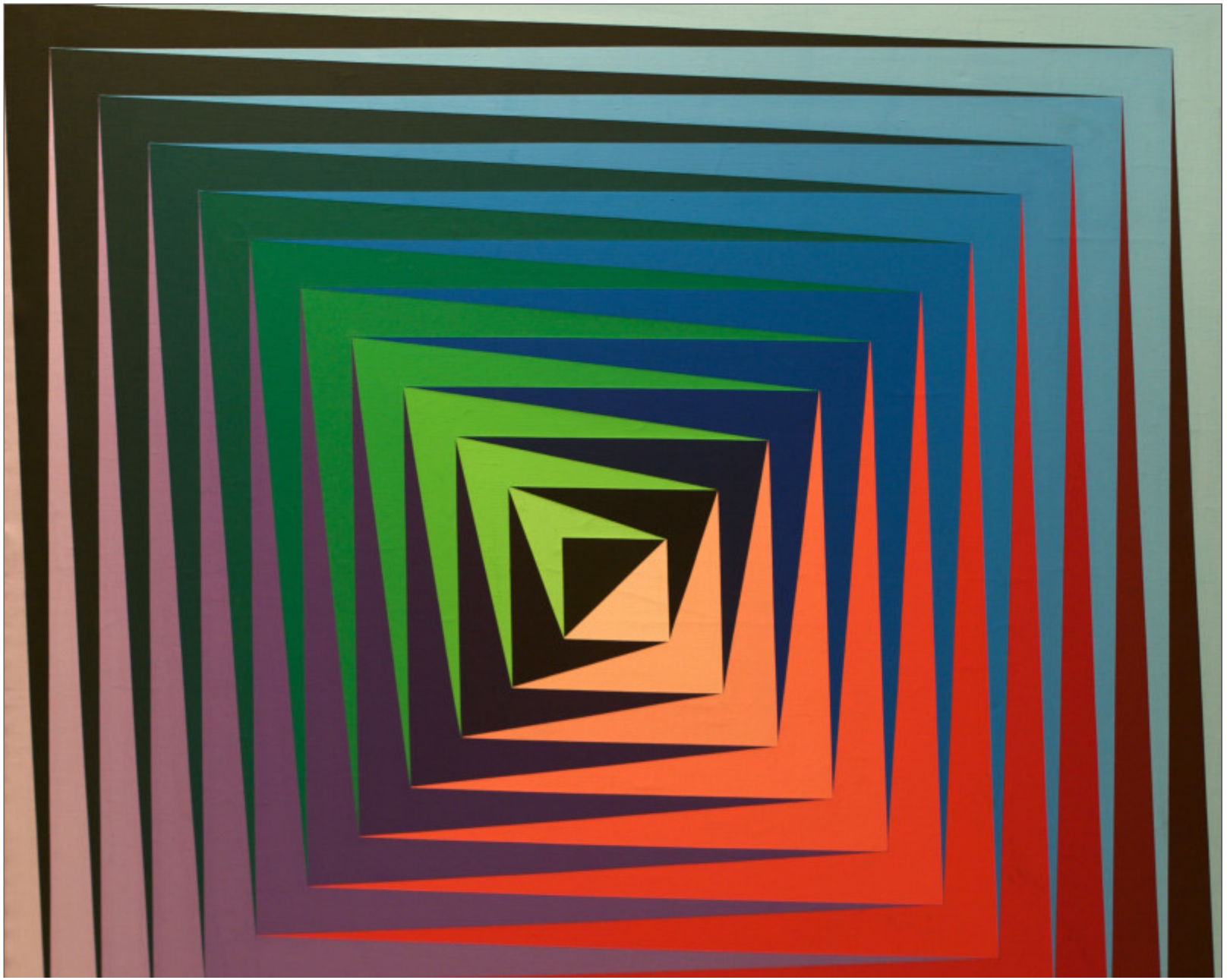






VICTOR VASARELY





3D

WEBGL

THREE.JS

```
var scene = new THREE.Scene();
var aspect = window.innerWidth / window.innerHeight;
//fov, aspect, near, far
var camera = new THREE.PerspectiveCamera( 75, aspect, 0.1, 1000 );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( 640, 480 );
document.body.appendChild( renderer.domElement );
```

FOV

```
var geometry = new THREE.BoxGeometry( 1, 1, 1 );
var material = new THREE.MeshNormalMaterial();
var cube = new THREE.Mesh( geometry, material );
scene.add( cube );
camera.position.z = 5;
```

```
var render = function () {
  requestAnimationFrame( render );
  cube.rotation.x += 0.01;
  cube.rotation.y += 0.01;
  renderer.render( scene, camera );
};

render();
```

TRANSFORMACIJE

```
cube.position.x = 2;
cube.position.set(2,3,4);
cube.rotation.x = 2;
cube.scale.set(2,3,4);
```

OBJEKTI

```
new THREE.SphereGeometry( 1, 5, 5);
new THREE.TorusGeometry( 1, 0.2, 10, 100);
new THREE.BoxGeometry(  1, 1, 1 );
new THREE.ConeGeometry( 12, 10,10);
```

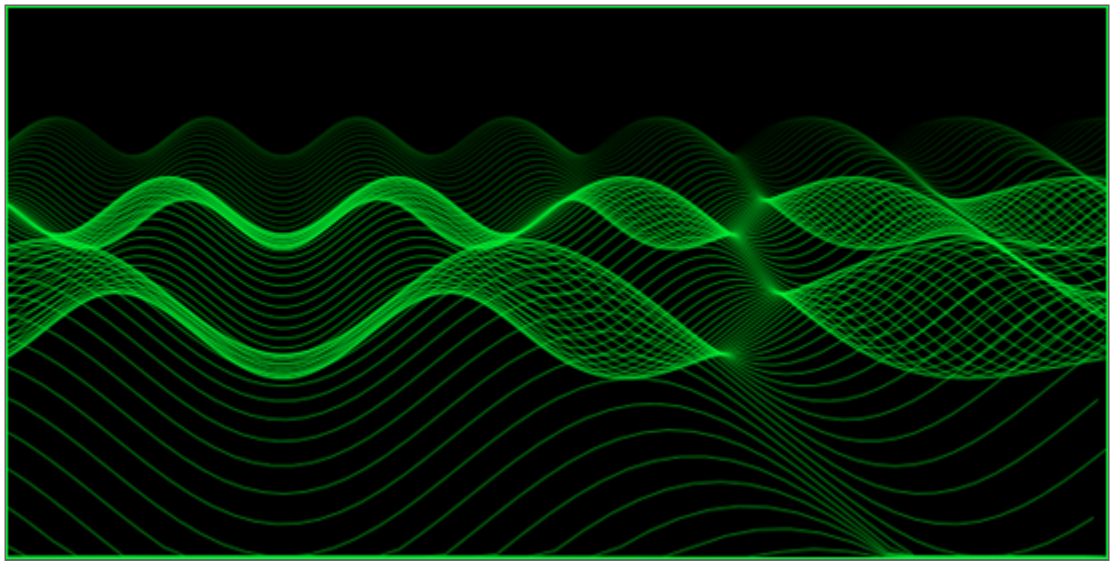
MATERIJALI

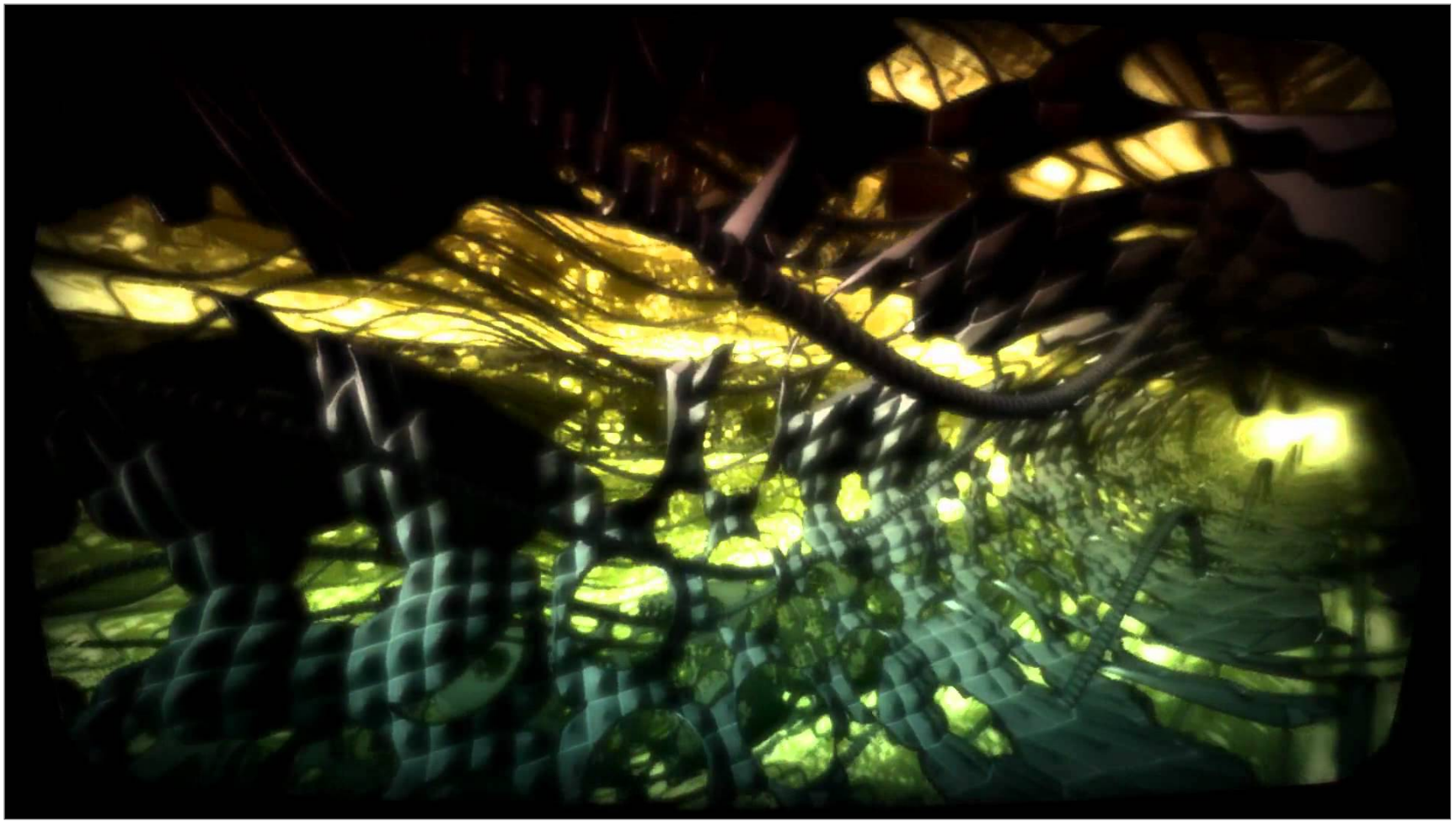
```
var normal = new THREE.MeshNormalMaterial({
});
var basic  = new THREE.MeshBasicMaterial({
    color: 0xff2d3d,
    wireframe: true
});
var phong  = new THREE.MeshPhongMaterial({
    color: 0xff2d3d,
    specular:0xffaaff,
    shininess: 20
});
var lambert = new THREE.MeshLambertMaterial({
    color: 0x11aabb
});
```

OSVETLJENJE

```
new THREE.DirectionalLight( 0xffffff, 1.0);
new THREE.AmbientLight( 0x404040 );
new THREE.PointLight(  0xff0000, 1, 100);
new THREE.SpotLight(  0xff0000, 1, 100);
```

DEMOSCENE





- **kosmoplovci**
- **pouet.net**
- **we are back**

GAME DEVELOPMENT

- **Babylon3d**
- **Phaser**
- **Unity3d**
- **Unreal Engine**

GAME UP

18. DOM OMALDINE