Interfaces de usuario e Interacción

Formas, Eventos de Usuarios (y POO)

Clases y objetos [Vista Gral.]

Clases y objetos [Vista Gral.]

Arrancar, Acelerar, Frenar, Girar Izq, Girar Derecha

- Girar (Sentido);

Clases y objetos [Vista Gral.]

- Automóvil → Es el objeto
- Propiedades: Características (Marca, Modelo, Color)
- Métodos: Acciones que puede realizar (frenar, arrancar, etc)

Y la clase?

- Una clase es “una plantilla” o modelo que define al objeto.
- La *clase* ***es la idea genérica*** que se tiene de **“todos”** los automóviles

Clases y objetos [Vista Gral.]

Ford Mustang Boss 302

Ford Mustang GT 350

Ford Mustang Mach1

’60 – ‘70

Ford Mustang GT 500

2017

Clases y objetos [Vista Gral.]

- ***Clase*:** definiciones de las propiedades y comportamiento de un “algo”.

- ***Objeto*:** una instancia de una clase.

- ***Método*:** Algoritmo asociado a un objeto.

- ***Propiedad o atributo*:** datos asociados a un objeto.

JS & POO

```
class Figure {
    constructor(posX, posY, fill, context) {
        this.posX = posX;
        this.posY = posY;
        this.fill = fill;
        this.context = context;
    }

    setFill(fill) {
        this.fill = fill;
    }

    getPosition() {
        return {
            x: this.getPosX(),
            y: this.getPosY()
        };
    }

    getPosX() {
        return this.posX;
    }
    getPosY() {
        return this.posY;
    }
    getFill() {
        return this.fill;
    }

    draw() {
        this.context.fillStyle = this.fill;
    }
}
```

JS & POO

```
class Circle extends Figure {
    constructor(posX, posY, radius, fill, context) {
        super(posX, posY, fill, context);

        this.radius = radius;
    }

    draw() {
        super.draw();
        this.context.beginPath();
        this.context.arc(this.posX, this.posY, this.radius, 0, 2 * Math.PI);
        this.context.fill();
        this.context.closePath();
    }

    getRadius() {
        return this.radius;
    }
}
```

JS & POO

```
class Rect extends Figure {
```

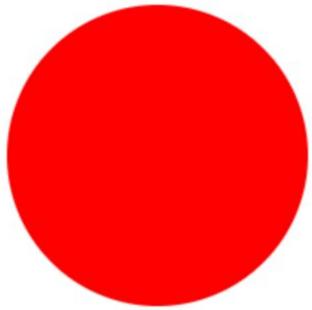
```
draw() {
    super.draw();
    this.context.fillRect(this.posX, this.posY, this.width, this.height);
}
```

Aplicar lo que sabemos de Context 2D:

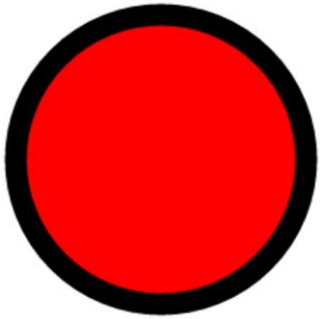
- ***Líneas***
- ***Rectángulo***
- ***Imágenes***
- ***Texto***
- ***Setear colores***
- ***Setear grosor y estilo de lápiz***
- ***…***

Dibujar un Círculo en el context

```
var ctx = document.getElementById("canvas").getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.beginPath();
ctx.arc(250, 250, 100, 0, Math.PI * 2);
ctx.fill();
ctx.closePath();
```



```
ctx.beginPath();
ctx.arc(250, 250, 100, 0, Math.PI * 2);
ctx.lineWidth = 15;
ctx.lineCap = 'round';
ctx.strokeStyle = 'black';
ctx.stroke();
ctx.closePath();
```



Fill styles

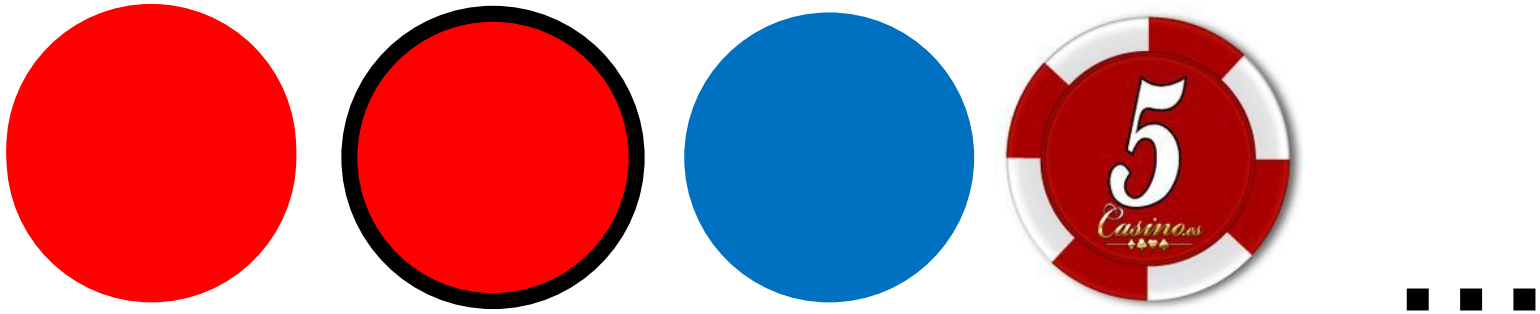
Gradiente

```
var gradient = ctx.createLinearGradient(10, 90, 200, 90);
gradient.addColorStop(0, 'black');
gradient.addColorStop(1, 'white');
ctx.fillStyle = gradient;
ctx.fillRect(10, 10, 200, 250);
```

Imagen

```
var image = ctx.createPattern(img,"repeat");
ctx.rect(0, 0, 150, 100);
ctx.fillStyle = image;
ctx.fill();
```

Dibujar un Círculo en el context



...

Eventos

- Mecanismos que permiten crear una interacción entre los elementos del documento y a las acciones del usuario.

- Existe un número de eventos definidos para todos los elementos:
  - *ONLOAD*
  - *ONMOUSEDOWN*
  - *ONMOUSEENTER*
  - *ONMOUSEDOWN*
  - *…..*

http://www.htmlquick.com/es/reference/events.html
http://www.javascripture.com/MouseEvent


4 EN LINEA
JUEGO DE ESTRATEGIA
Rulibal
4 EN LINEA
JUEGO DE ESTRATEGIA
Rulibal

Ensayo…

- ***Ficha***
- ***Tablero***

- ***Ficha: Color***
- ***Dibujarse, moverse, etc***

Más detalles, próxima clase…