

Amalgama. Treball Final de Grau.

Grau de Multimèdia. Creació Gràfica. Desembre 2016.

/ Memòria. Annex 2.

Codi font [extractes].

Autor: Benjamí Vilardell i Pareta

Consultor: Ivan Serrano Regol

Professor: Irma Vilà Òdena



Índex.

1. Visió general.	2
2. Funcions principals.	3
Posicionament d'un botó.	3
Funció windowResized.	3
Funció mousePressed.	3
Funció keyPressed.	4
3. Funcions Botons.	5
Funció crearBotoColor.	5
Funció crearBotoAltaveu.	5
4. Funcions text.	6
Funció alçadaText.	6
5. Funcions Imatges.	7
Funció crearImatgeCercle.	7
Funció crearImatgeSencera.	7
Funció actualitzarImatge.	8
Funció crearImatge.	8

1. Visió general.

Davant l'obvietat que tot el codi és necessari per a una reproducció òptima de la pàgina web, i tenint en compte que per la filosofia pròpia d'aquest projecte no s'han creat elements sobrers que no tinguessin un sentit i una utilitat clara d'aportació significativa, aquí només se'n destaquen aquelles porcions de codi més importants, ja sigui per la seva lògica imprescindible o per les seves pròpies característiques de resolució d'un problema o de capacitat estructural.

A continuació, doncs, es llisten aquells extractes de codi més destacables dels diferents arxius que conformen l'estructura de la pàgina web.

2. Funcions principals.

En l'arxiu Funcions Principals, s'hi encapsulen bona part de les funcions estructurals de la pàgina web.

Posicionament d'un botó.

Crea un botó amb unes característiques o altres segons si el ratolí hi és damunt. En aquest cas, l'exemple que crea el botó amb el text "-A".

```
if(sobreCercle(posicioBotoMenysFontX,posicioBotoMenysFontY,diametreBoto)){
    missatgeBoto("disminuir lletra");
    cursor(HAND);
    crearBotoText("-A",posicioBotoMenysFontX,posicioBotoMenysFontY,diametreBoto,colorTracPrincipal,colorFonsLlenc);
    sobreBotoMenysFont=true;
}else{
    crearBotoText("-A",posicioBotoMenysFontX,posicioBotoMenysFontY,diametreBoto,colorTracPrincipal,colorFonsLlenc,colorTracPrincipal);
    sobreBotoMenysFont=false;
}
```

Funció windowResized.

S'hi actualitzen les mides del llenç i de les posicions dels elements en el cas d'un redimensionament de la pantalla.

```
function windowResized() {
    actualitzarLlenc(windowWidth, windowHeight);
    if(!esMobil()){
        detectarMidaPantalla();
    }
    actualitzarPosicions();
}
```

Funció mousePressed.

S'hi encapsulen les funcions a executar en el cas de que es premi el ratolí.

```
function mousePressed() {
    textFont(fontPrincipal);
    if(sobreImatgeActual){imatgeResultat=crearImatgeSencera(imatgeCercle); imatgeAcabada=true;}
    (...)
}
```

2. Funcions principals.

Funció keyPressed.

S'hi encapsulen les funcions a executar en el cas que es pulsi una tecla.

```
function keyPressed(){
    textFont(fontPrincipal);
    if(key == "c" || key == "C"){ // si es pulsa lletra c
        if(mostrarSectorCites){mostrarSectorCites = false;}
    }else{mostrarSectorCites=true;}
    (...)
}
```

3. Funcions Botons.

De l'arxiu Funcions Botons, se'n destaca el següent codi.

Funció crearBotoColor.

Creació d'un botó amb dues zones de color.

```
function crearBotoColor(x,y,amplada,color1,color2,color3){  
  ellipseMode(CENTER);  
  stroke(color1).strokeWeight(1).fill(color3);  
  ellipse(x,y,amplada);  
  noStroke().strokeWeight(1).fill(color2);  
  ellipse(x,y,amplada-10);  
}
```

Funció crearBotoAltaveu.

Exemple de la creació d'un botó amb el seu propi dibuix.

```
function crearBotoAltaveu(x,y,amplada,color1,color2,color3){  
  ellipseMode(CENTER);  
  stroke(color1).strokeWeight(1).fill(color2);  
  ellipse(x,y,amplada);  
  stroke(color3).strokeWeight(1);  
  line(x-(amplada/4), y-(amplada/8), x-(  
    amplada/4), y+amplada/8);  
  line(x, y-(amplada/4), x, y+amplada/4);  
  line(x-(amplada/4), y-(amplada/8), x, y-(amplada/4));  
  line(x-(amplada/4), y+amplada/8, x, y+amplada/4);  
  line(x+(amplada/6), y-(amplada/8), x+(amplada/6), y+(ampla  
    da/8));  
  line(x+(amplada/4), y-(amplada/4), x+(amplada/4), y+(ampla  
    da/4));  
  if(musicaActiva){line(x-(amplada/4), y-(amplada/4), x+(ampla  
    da/4), y+(amplada/4))}  
}
```

4. Funcions text.

La funció de l'arxiu Funcions Text és la següent, imprescindible per calcular la llargada dels camps de text, en P5, adaptant-se a diferents canvis d'amplada de pantalla.

Funció alçadaText.

Funció per calcular els píxels que ocupa una cadena de text segons l'amplada de la pantalla en un moment donat. Aquesta funció seria innecessària en altres llenguatges de programació.

```
function alçadaText(escrit, amplada){
  var words = escrit.split(' ');
  var linia = '';
  var h = textLeading();
  for (var i = 0; i < words.length; i++) {
    var testLinia = linia + words[i] + ' ';
    var testAmplada = drawingContext.measureText
      (testLinia).width;
    if (testAmplada > amplada && i > 0) {
      linia = words[i] + ' ';
      h += textLeading();
    } else {
      linia = testLinia;
    }
  }
  return h;
}
```

5. Funcions Imatges.

Com a projecte gràfic que és, en l'arxiu Funcions Imatges és on hi haurà algunes de les funcions més significatives a nivell expressiu.

Funció crearImatgeCercle.

Funció que converteix una imatge en una imatge circular, tot escollint els píxels del cercle central de la imatge rebuda.

```
function crearImatgeCercle(imatgeOrigen){
    var imatgeCreada = createImage(imatgeOrigen.width, imatgeOrigen.
height);
    imatgeCreada.loadPixels();
    var distanciaMinima = min(imatgeOrigen.width, imatgeOrigen.
height);
    for (var x = 0; x < imatgeOrigen.width; x++) {
        for (var y = 0; y < imatgeOrigen.height; y++) {
            var coloret = imatgeOrigen.get(x,y);
            if(puntSobreCercle(x, y, imatgeOrigen.width/2, imatgeOrigen.
height/2, distanciaMinima)){
                imatgeCreada.set(x, y, coloret);
            }else{imatgeCreada.set(x, y, colorFonsLlenc); }
        }
    }
    imatgeCreada.updatePixels();
    return imatgeCreada;
}
```

Funció crearImatgeSencera.

Funció per crear una imatge, píxel a píxel, tal com és en origen.

```
function crearImatgeSencera(imatgeOrigen){
    var imatgeCreada = createImage(imatgeOrigen.width, imatgeOrigen.
height);
    imatgeCreada.loadPixels();
    for (var x = 0; x < imatgeOrigen.width; x++) {
        for (var y = 0; y < imatgeOrigen.height; y++ ) {
            var coloret = imatgeOrigen.get(x,y);
            imatgeCreada.set(x, y, coloret);
        }
    }
    imatgeCreada.updatePixels();
    imatgeAcabada = true;
    return imatgeCreada;
}
```


5. Funcions Imatges.

Funció actualitzarImatge.

Funció que crea una imatge, píxel a píxel, a cada volta del draw(). Se'n pot configurar el nombre màxim de píxels, els píxels que crea, la distància entre píxels que es crea de manera atzarosa i es pot escollir si després de crear el màxim de píxels configurats es vol crear la imatge sencera o mostrar-la parcialment segons la part creada.

```
function actualitzarImatge(imatgeOrigen){
    if(comptadorVoltes < numeroMaximVoltes && !sobreImatgeActual &&
    !esMobil()){
        frameRate(frA);
        var diferencia = 3;
        var punt;
        var quantitatPunts = 50;
        for (punt = 0; punt < quantitatPunts; punt++) {
            puntX = puntX + int(random(-diferencia, diferencia));
            while (puntX < 0 || puntX > imatgeOrigen.width-1){puntX =
puntX + int(random(-diferencia, diferencia));}
            puntY = puntY + int(random(-diferencia, diferencia));
            while (puntY < 0 || puntY > imatgeOrigen.height-1){puntY =
puntY + int(random(-diferencia, diferencia));}

            var coloret = imatgeOrigen.get(puntX,puntY);
            imatgeCreada.set(puntX, puntY, coloret);
            quantitatPunts += 50;
        }

        imatgeCreada.updatePixels();
        frameRate(frN);
        comptadorVoltes++;
    }
    if(comptadorVoltes >= numeroMaximVoltes){imatgeAcabada =
true;}
    return imatgeCreada;
}
```

Funció crearImatge.

Funció que caldrà haver executat abans de la funció crearImatge per a inicialitzar-ne les característiques que es necessitaran.

```
function crearImatge(imatgeOrigen){
    imatgeCreada = createImage(imatgeOrigen.width, imatgeOrigen.
height);
    imatgeCreada.loadPixels();
    puntX = random(imatgeOrigen.width);
    puntY = random(imatgeOrigen.height);
}
```

