

Integració Digital de Continguts

Processing

PRA 2
Pere Amengual Gomila
Octubre 2015

Exercici 1

Carregar una imatge de fons que es mantindrà estàtica durant l'execució del programa.

Per realitzar aquest exercici només és necessari declarar un objecte *bg* del tipus *PImage*. En el *setup()* definir les mides de pantalla, que coincideixen amb les de la imatge. Després inicialitzem el valor de la imatge, assignant-li un fitxer *paissatge.png* que prèviament hem situat a la carpeta */data*.

Per acabar, i amb la finalitat que la imatge s'imprimeixi com a fons a cada quadre, dins *draw()* l'assignem al background.

```
PImage bg;  
  
void setup() {  
  size(800, 600);  
  bg = loadImage("paissatge.png");  
}  
  
void draw() {  
  background(bg);  
}
```

Exercici 2

(4 punts) Definir un guió format per 2 frames que compleixi amb aquestes condicions:

1. Frame 1: apareix un text per pantalla indicant que estem "PARATS".
2. Frame 2: apareix un text per pantalla indicant que estem "SALTANT".
3. Els textos anteriors s'han d'escriure des de funcions que crearem fora de la funció `draw()`.
4. La acció que ens farà passar d'un frame a un altre serà el clic del ratolí.

Per saber si estem en el frame 1 o en el frame 2 farem servir la variable booleana *saltant*, que prendrà el valor *false* pel frame 1 i *true* pel frame 2.

Dins el *setup()* les mides del text es defineixen amb *textSize(48)* i assignem a la variable *saltant* el valor *false*.

Dins *draw()* cridem a la funció *imprimirStatus()* que ens treurà per pantalla el text "SALTANT" o "PARATS" en funció del valor de la booleana *saltant*.

Per alternar el valor de la booleana feim servir la funció *mouseClicked()* que escolta els events de ratolí. Quan es clica el ratolí, *saltant* canvia al valor contrari.

```
PImage bg;
boolean saltant;

void setup() {
  size(800, 600);
  textSize(48);
  bg = loadImage("paisatge.png");
  saltant = false; // false equival a frame 1/parats i true
                  // equival a frame 2/saltant
}

void draw() {
  background(bg);
  imprimirStatus();
}

void imprimirStatus() {
  if (saltant) {
    text("SALTANT", 100, 100);
  } else {
    text("PARATS", 100, 100);
  }
}

void mouseClicked() {
  saltant = !saltant;
}
```

Exercici 3

(4 punts) A partir de les capes del fitxer "jumping.psd" crearem 3 imatges diferents. Al arrencar el programa estarem en el frame 1 i apareixerà una d'aquestes imatges. Quan anem al frame 2, utilitzarem les 3 imatges per anar-les intercanviant entre si i simular una animació del personatge saltant.

Si tornem al frame 1 el personatge quedarà quiet fins que tornem a saltar al frame 2.

Hem introduït un array d'imatges, la qual cosa ens permetrà accedir a elles fent servir estructures de control iteratiu i canviar l'sketch per obtenir un nombre arbitrari d'imatges per l'animació. Declarem la variable *contador*, que ens servirà per anar registrant el temps que la imatge està saltant, i la variable *velocitat*, que dividirà el valor anterior per obtenir un valor enter que augmentarà amb el ritme visual desitjat i que farem servir per canviar d'imatge.

Dins *setup()* definim una velocitat de quadre de 30 quadres per segon. Després fem servir una estructura for per carregar les imatges dins l'array. Situem el valor inicial de *saltant* a *false* per començar en el frame 1 (quiet). També posem el *contador* a zero i definim la *velocitat* amb el valor 10 (valors més baixos proporcionen velocitats més altes).

Dins *draw()* ara també cridem la funció *dibuixarPersonatge()* que ens imprimeix la corresponent dins el bucle i, a més, fa avançar la imatge a imprimir fent servir l'expressió $(\text{contador}/\text{velocitat})\% \text{personatge.length}$.

```
PImage bg;
```

```
PImage[] personatge = new PImage[3]; // podem canviar les mides del array per animar amb un nombre arbitrari d'imatges
boolean saltant;
int contador;
int velocitat;
```

```
void setup() {
    size(800, 600);
    frameRate(20);
    textSize(48);
    bg = loadImage("paisatge.png");
    for ( int i = 0; i < personatge.length; i++ ) {
        personatge [i] = loadImage( "personatge" + i + ".png" );
    }
    saltant = false; // false equival a frame 1/parats i true equival a frame 2/saltant
    contador = 0;
    velocitat = 10;
}
```

```
void draw() {
    background(bg);
    imprimirStatus();
    dibuixarPersonatge();
}
```

```
void imprimirStatus() {
    if (saltant) {
        text("SALTANT", 100, 100);
    }
}
```

Integració Digital de Continguts - PRA 2

```
    } else {  
        text("PARATS", 100, 100);  
    }  
}  
  
void dibuixarPersonatge() {  
    if (saltant) {  
        contador++;  
    }  
    image (personatge[(contador/velocitat) % personatge.  
length], 200, 300);  
}  
  
void mouseClicked() {  
    saltant = !saltant;  
}
```

Exercici 4

(2 punts) Per millorar el resultat podem editar l'imatge del personatge perquè el fons sigui transparent. També podem afegir algun element extra com per exemple un ocell que passi volant, es clar que aquest ocell no pot parar de volar estiguem al frame que estiguem (utilitzeu una funció específica per controlar-lo).

En aquesta darrera versió, les millores han consistit en afegir so de fons i també so per el salt de la corda, una animació constant amb dos elements (sol i lluna) que apareixen fent servir una màscara sobre el paisatge i un canvi de llum de l'escena segons es passa del dia a la nit.

En primer lloc declarem els objectes *comba* i *forest* que farem servir per la funcionalitat del so. Ara també tenim les imatges *sol*, *lluna* i *bg_mask*, que farem servir per simular l'efecte del sol i la lluna sortint per l'horitzó.

Degut a limitacions de la llibreria de so de Processing, he hagut de preparar variables addicionals per evitar la repetició innecessària dels sons ja que no es disposa de cap mètode de tipus *isPlaying()* ni tampoc sembla funcionar correctament el mètode *loop()*. La variable booleana *haSonat* ens ajudarà a impedir que es redispari el so de la comba a cada frame que la funció *dibuixarPersonatge()* imprimeix per pantalla la imatge que es correspon amb el moment de saltar. La variable *properBosc* ens ajuda a implementar la reproducció en bucle del so de fons dins la funció *sonsBosc()*.

Per millorar la llegibilitat del codi, la variable *salt* emmagatzemarà

quina imatge del personatge és la actual.

Per crear l'efecte de màscara en el que el sol surt per l'horitzó, dins *draw()* imprimim primer el paisatge sencer, després el sol i la lluna, després el paisatge amb la màscara per no imprimir el cel i, finalment, el personatge que salta. Posteriorment, amb un rectangle d'opacitat variable enfosquim tota la escena.

Les funcions *dibuixarSol()*, *dibuixarLluna()* i *dibuixarNit()* reben el paràmetre del frame actual obtingut mitjançant la suma de *frameCount* i un offset, o desplaçament, assignat en el moment de la crida. Així podem fer que el sol, la lluna i la foscor apareguin en moments distints del cicle dia/nit.

El nivell de foscor depen d'un càlcul trigonomètric amb els valors convenientment escalats i desplaçats. També les posicions del sol i la lluna fan servir funcions trigonomètriques per obtenir una trajectòria circular.

A l'hora de dibuixar el personatge, ha estat important tenir en compte que el so de la corda no podia redisparar-se durant una mateixa posició, motiu pel qual només es reproduïx el so de la corda si el valor de la booleana *haSaltat* és fals. A cada iteració del salt, es canvia lleugerament i de forma aleatòria la velocitat de reproducció del salt fent servir *comba.rate(random(0.8, 1.1))*, amb la finalitat d'obtenir un so més variat i realista.

Integració Digital de Continguts - PRA 2

Els antecedents dels mitjans interactius

La funció *sonsBosc()* fa servir la variable *properBosc* per emmagatzemar el moment en que ha de fer sonar el so de fons. En el moment exacte en que comença una reproducció actualitzem el valor de la variable sumant-li $int(forest.duration()*1000)-100$ i així ens assegurem que el proper so es tornarà a reproduir en el moment just abans que s'acabi de reproduir el so actual.

```
import processing.sound.*;

SoundFile comba, forest;

PImage paisatge, bg, bg_mask, sol, lluna;
PImage[] personatge = new PImage[3]; // podem canviar
les mides del array per animar amb u nombre arbitrari
d'imatges
boolean saltant, haSonat;
int contador, velocitat_salts, salt, properBosc;

void setup() {
    size(800, 600);
    frameRate(30);
    textSize(48);
    paisatge = loadImage("paisatge.png");
    bg = loadImage("paisatge.png");
    bg_mask = loadImage("paisatge_mask.png");
    sol = loadImage("sun.png");
    lluna = loadImage("moon.png");
    bg.mask(bg_mask);
```

```
    for ( int i = 0; i< personatge.length; i++ ) {
        personatge [i] = loadImage( "personatge" + i + ".png"
    );
    }
    saltant = false; // false equival a frame 1/parats i true
equival a frame 2/saltant
    contador = 0;
    properBosc = 0;
    velocitat_salts = 10;
    comba = new SoundFile(this, "comba.mp3");
    forest = new SoundFile(this, "forest.mp3");
}

void draw() {
    background(paisatge);
    dibuixarSol(frameCount+600);
    dibuixarLluna(frameCount+1200);
    image (bg, 0, 0);
    imprimirStatus();
    dibuixarPersonatge();
    dibuixarNit(frameCount+700);
    sonsBosc();
}

void imprimirStatus() {
    fill (255, 255, 255);
    if (saltant) {
        text("SALTANT", 50, 50);
    } else {
        text("PARATS", 50, 50);
    }
}
```

Integració Digital de Continguts - PRA 2

```
void dibuixarNit(int pos) {
    float y = sin((pos)/200.0)*65+55;
    if (true) {
        fill (0, 0, 40, y);
        rect (0, 0, width, height);
    }
}

void dibuixarSol (int pos) {
    float x = cos((pos)/200.0)*420+420;
    float y = int(sin((pos)/200.0)*400+400);
    fill (255, 255, 255);
    image (sol, x, y, 150, 127);
}

void dibuixarLluna(int pos) {
    float x = cos((pos)/200.0)*420+420;
    float y = int(sin((pos)/200.0)*400+400);
    fill (255, 255, 255);
    image (lluna, x, y, 150, 127);
}

void dibuixarPersonatge() {
    if (saltant) {
        contador++;
    }
    salt = (contador/velocitat_salts) % personatge.length;
    image (personatge[salt], 200, 300);
    if ( salt == 2) {
        if (!haSonat) {
            comba.play();

            comba.rate(random(0.8, 1.1));
            haSonat = true;
        }
    } else {
        haSonat = false;
    }
}

void sonsBosc() {
    // funció implementada degut a que no funciona
    // correctament el mètode loop()
    if (millis() > properBosc) {
        forest.play();
        properBosc = millis() + int(forest.
        duration()*1000)-100;
    }
}

void mouseClicked() {
    saltant = !saltant;
}
```

Bibliografia i fonts consultades

Imatges i sons addicionals

Sol. Autor desconegut Clipart gratuït. [en línia] http://images.clipartpanda.com/sun-clipart-png-sun_icon-7587cced3d0c975180e86393e3c-6cc9e.png

Lluna. Autor desconegut. Clipart gratuït. [en línia] <http://cdn.hdwallpaperhdpictures.in/Clipart/moon/moon-clipart-7.jpg>

Paisatge. Night to day - Farm time lapse. Jimmi Selix. [en línia] <https://www.youtube.com/watch?v=kQM6Q9Axyx0>

Bosc. Gamjorrr. countrysideambiance.wav [en línia] <https://www.freesound.org/people/gmajorrr/sounds/132719/>

Corda. Flowfx. Rope Spin i air like lasse version 3. [en línia] <http://www.freesfx.co.uk/sfx/rope>