import gifAnimation.\*;

import processing.video.\*;

import java.util.\*;

enum ActType{

PICTURE,

VIDEO,

PROGRAM

}

ArrayList<Act> PlayList;

int curActIdx = -1;

boolean isBlack = true;

boolean KBLock = false;

//SPACE: play PAUSE:z STOP:x BLINK:b

void settings(){

fullScreen();

}

void setup(){

background(40);

PlayList = new ArrayList<Act>();

LoadPlayList(sketchPath("media/PlayList.txt"));

curActIdx = PlayList.size() - 1;

println("Num of Act: " + PlayList.size());

for(int i = 0; i < PlayList.size(); i++){

println(i + ": " + PlayList.get(i).type);

}

}

void draw(){

imageMode(CORNER);

if(isBlack){

background(0);

}

else{

showImage(PlayList.get(curActIdx));

}

}

void keyPressed(){

if(!KBLock){

KBLock = true;

if(key == CODED){

if(keyCode == DOWN){

if(isBlack){

curActIdx++;

curActIdx %= PlayList.size();

play(PlayList.get(curActIdx));

isBlack = false;

}

else{

stop(PlayList.get(curActIdx));

isBlack = true;

}

}

else if(keyCode == UP){

if(isBlack){

play(PlayList.get(curActIdx));

isBlack = false;

}

else{

stop(PlayList.get(curActIdx));

if(curActIdx != 0){

curActIdx--;

}

else{

curActIdx = PlayList.size()-1;

}

isBlack = true;

}

}

else if(keyCode == RIGHT){

if(!isBlack){

play(PlayList.get((curActIdx+1) % PlayList.size()));

stop(PlayList.get(curActIdx));

curActIdx++;

curActIdx %= PlayList.size();

}

}

else if(keyCode == LEFT){

if(!isBlack){

int tmp = curActIdx-1;

if(tmp == -1) tmp = PlayList.size() - 1;

play(PlayList.get(tmp));

stop(PlayList.get(curActIdx));

curActIdx = tmp;

}

}

}

KBLock = false;

}

}

void movieEvent(Movie m) {

m.read();

}

void play(Act curAct){

Picture curPic;

if(curAct.type == ActType.PICTURE){

for(int i = 0; i < curAct.aPicture.size(); i++){

curPic = curAct.aPicture.get(i);

if(curPic.isAnimation){

if(curPic.isLoop) curPic.GIF.loop();

else curPic.GIF.play();

}

}

}

else if(curAct.type == ActType.VIDEO){

if(curAct.video.isLoop) curAct.video.movie.loop();

else curAct.video.movie.play();

}

else if((curAct.type == ActType.PROGRAM)){

curAct.program.run();

}

}

void stop(Act curAct){

Picture curPic;

if(curAct.type == ActType.PICTURE){

for(int i = 0; i < curAct.aPicture.size(); i++){

curPic = curAct.aPicture.get(i);

if(curPic.isAnimation) curPic.GIF.stop();

}

}

else if(curAct.type == ActType.VIDEO){

curAct.video.movie.stop();

}

}

void showImage(Act curAct){

Picture curPic;

if(curAct.type == ActType.PICTURE){

for(int i = 0; i < curAct.aPicture.size(); i++){

curPic = curAct.aPicture.get(i);

if(curPic.isAnimation){

image(curPic.GIF, curPic.x, curPic.y, curPic.w, curPic.h);

}

else image(curPic.Image, curPic.x, curPic.y, curPic.w, curPic.h);

}

}

else if(curAct.type == ActType.VIDEO){

image(curAct.video.movie, 0, 0, width, height);

}

else if((curAct.type == ActType.PROGRAM)){

curAct.program.draw();

}

}

void LoadPlayList(String file){

String lines[] = loadStrings(file);

String[] buf, buf2;

ArrayList<PlayListBlock> aPLB = new ArrayList<PlayListBlock>(); //record <Act, lines> in the PlayList.txt

for (int i = 0 ; i < lines.length; i++) {

buf = match(lines[i], "(.\*)//.\*");

if(buf != null){

lines[i] = buf[1];

}

buf = match(lines[i], "([1-9][0-9]\*):");

if(buf != null){

aPLB.add(new PlayListBlock(Integer.parseInt(buf[1]), i));

}

}

Collections.sort(aPLB, new Comparator<PlayListBlock>() {

@Override

public int compare(PlayListBlock PLB1, PlayListBlock PLB2)

{

return ((Integer)PLB1.BeginLine).compareTo(((Integer)PLB2.BeginLine));

}

});

for(int i = 0; i < aPLB.size()-1; i++){

aPLB.get(i).EndLine = aPLB.get(i+1).BeginLine-1;

}

aPLB.get(aPLB.size()-1).EndLine = lines.length-1;

Collections.sort(aPLB, new Comparator<PlayListBlock>() {

@Override

public int compare(PlayListBlock PLB1, PlayListBlock PLB2)

{

return ((Integer)PLB1.ActNo).compareTo(((Integer)PLB2.ActNo));

}

});

for(int i = 0; i < aPLB.size(); i++){

for(int j = aPLB.get(i).BeginLine+1; j <= aPLB.get(i).EndLine; j++){

buf = splitTokens(lines[j], ", ");

if(buf.length != 0){

println("(i,j): (" + i + ", " + j + ") " + buf.length);

buf2 = match(lines[j], "(.+)\\.([^\\. ]+)");

if(buf2 != null){

buf2[2] = buf2[2].toLowerCase();

println("(i,j): (" + i + ", " + j + ") " + buf2[2]);

if(buf2[2].equals("jpg") || buf2[2].equals("gif") || buf2[2].equals("tga") || buf2[2].equals("png")){

println("(i,j): (" + i + ", " + j + ") " + "Infif");

while(PlayList.size() < aPLB.get(i).ActNo) PlayList.add(new Act(ActType.PICTURE));

if(buf2[2].equals("gif")){

switch(buf.length){

case 1:

PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0])));

break;

case 2:

if(Integer.parseInt(buf[1]) == 0) PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), false));

else PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), true));

break;

case 3:

PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2])));

break;

case 4:

if(Integer.parseInt(buf[3]) == 0) PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2]), false));

else PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2]), true));

break;

case 5:

PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2]), Integer.parseInt(buf[3]), Integer.parseInt(buf[4])));

break;

case 6:

if(Integer.parseInt(buf[5]) == 0) PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2]), Integer.parseInt(buf[3]), Integer.parseInt(buf[4]), false));

else PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(this, sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2]), Integer.parseInt(buf[3]), Integer.parseInt(buf[4]), true));

break;

}

}

else{

switch(buf.length){

case 1:

PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(sketchPath("media/" + buf2[0])));

break;

case 3:

PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2])));

break;

case 5:

PlayList.get(aPLB.get(i).ActNo-1).aPicture.add(new Picture(sketchPath("media/" + buf2[0]), Integer.parseInt(buf[1]), Integer.parseInt(buf[2]), Integer.parseInt(buf[3]), Integer.parseInt(buf[4])));

break;

}

}

}

if(buf2[2].equals("mov") || buf2[2].equals("mp4")){

while(PlayList.size() < aPLB.get(i).ActNo) PlayList.add(new Act(ActType.VIDEO));

switch(buf.length){

case 1:

PlayList.get(aPLB.get(i).ActNo-1).video = new Video(this, sketchPath("media/" + buf2[0]));

break;

case 2:

if(Integer.parseInt(buf[1]) == 0) PlayList.get(aPLB.get(i).ActNo-1).video = new Video(this, sketchPath("media/" + buf2[0]), false);

else PlayList.get(aPLB.get(i).ActNo-1).video = new Video(this, sketchPath("media/" + buf2[0]), true);

break;

}

}

if(buf2[2].equals("prog")){

while(PlayList.size() < aPLB.get(i).ActNo) PlayList.add(new Act(ActType.PROGRAM));

if(buf2[1].equals("movingShadow")) PlayList.get(aPLB.get(i).ActNo-1).program = new movingShadow(this);

}

}

}

}

}

}

class Act{

Act(ActType type)

{

if(type == ActType.PICTURE){

aPicture = new ArrayList<Picture>();

}

this.type = type;

}

public ActType type;

public ArrayList<Picture> aPicture = null;

public Video video;

public Program program;

}

class Picture{

Picture(PApplet parent, String file, int x, int y, int w, int h, boolean isLoop)

{

this.GIF = new Gif(parent, file);

println("GIF in PIC");

isAnimation = true;

this.x = x;

this.y = y;

GifDecoder d = new GifDecoder();

d.read(file);

if(w == 0) this.w = d.getFrameSize().width;

else this.w = w;

if(h == 0) this.h = d.getFrameSize().height;

else this.h = h;

this.isLoop = isLoop;

GIF.ignoreRepeat();

}

Picture(PApplet parent, String file, int x, int y, int w, int h)

{

this(parent, file, x, y, w, h, false);

}

Picture(PApplet parent, String file, int x, int y, boolean isLoop)

{

this(parent, file, x, y, 0, 0, isLoop);

this.w = width;

this.h = height;

}

Picture(PApplet parent, String file, int x, int y)

{

this(parent, file, x, y, false);

}

Picture(PApplet parent, String file, boolean isLoop)

{

this(parent, file, 0, 0, isLoop);

}

Picture(PApplet parent, String file)

{

this(parent, file, 0, 0, false);

}

Picture(String file, int x, int y, int w, int h){

Image = loadImage(file);

isAnimation = false;

this.x = x;

this.y = y;

if(w == 0) w = Image.width;

else this.w = w;

if(h == 0) h = Image.height;

else this.h = h;

}

Picture(String file, int x, int y){

this(file, x, y, 0, 0);

w = width;

h = height;

}

Picture(String file){

this(file, 0, 0);

}

boolean isAnimation;

Gif GIF;

PImage Image;

int x,y;

int w,h;

boolean isLoop;

}

class Video{

Video(PApplet parent, String file, boolean isLoop){

movie = new Movie(parent, file);

this.isLoop = isLoop;

}

Video(PApplet parent, String file){

this(parent, file, false);

}

Movie movie;

boolean isLoop;

}

class PlayListBlock{

PlayListBlock(int ActNo, int BeginLine){

this.ActNo = ActNo;

this.BeginLine = BeginLine;

}

int ActNo;

int BeginLine;

int EndLine;

}

abstract class Program{

public abstract void run();

public abstract void stop();

public abstract void draw();

}

class movingShadow extends Program{

private Movie movie;

private Gif GIF;

private int w\_GIF, h\_GIF;

private int x\_GIF, y\_GIF;

private int moveSpeed = 30;

public movingShadow(PApplet parent){

movie = new Movie(parent, sketchPath("media/battle.mov"));

GIF = new Gif(parent, sketchPath("media/shadow.gif"));

GifDecoder d = new GifDecoder();

d.read(sketchPath("media/shadow.gif"));

w\_GIF = d.getFrameSize().width;

h\_GIF = d.getFrameSize().height;

x\_GIF = 0;

y\_GIF = height - h\_GIF;

}

public void run(){

GIF.ignoreRepeat();

GIF.loop();

movie.loop();

}

public void stop(){

GIF.stop();

movie.stop();

}

public void draw(){

x\_GIF = mouseX;

y\_GIF = mouseY;

image(movie,0,0,width,height);

blend(GIF, 0, 0, w\_GIF, h\_GIF, x\_GIF, y\_GIF, w\_GIF, h\_GIF, LIGHTEST);

}

}