# Gliding in the Sky

## Artist Statement

In my final project, I use a couple images to present a glider flying in the sky. I choose 4 landscape images as the background and edit another 2glider images with a transparent background. By using Processing, I let the glider image float on the background images. The images are loaded and displayed by using PImage.

You can use the follow keys to make some interactions in the window:

* ‘A’ or ‘a’ – move the glider faster
* ‘S’ or ‘s’ – slow down the glider down
* LEFT, RIGHT, UP and DOWN arrow keys – change directions

By applying the sound library I learned in this course, the glider plays the different sounds (made by using Noise and Chirp in Audacity) when the glider changes its direction whenever by the keystrokes, or it hits the boundaries of the window, and it accelerates or slows down.

## References:

I tried to upload my work to Openprocessing.com but it does not support the script with libraries – probably Minim sound library. And, ePortfolio has been under maintenance. I will upload the URL as soon as the site is up. In the meantime, you can download the source code from Github, <https://github.com/jonliu6/ArtWork/blob/master/Digital/final_project_gliding_in_the_sky.zip?raw=true>

## Screenshots:D:\MyWork\ArtCourse\Processing\FinalProject\screen1.PNGD:\MyWork\ArtCourse\Processing\FinalProject\screen2.PNG



## Source Code:

// This is the final project of Art of Computing. It displays a random image (out of 4) as background and moved a glider image around.

// User can use the following keys to change the direction and speed of the glider:

// - A or a to accelerate

// - S or s to slow down

// - LEFT, RIGHT, UP and DOWN arrow keys to control the direction

// When the glider is outside the main window, it will automatically change the direction with a chrip sound.

import ddf.minim.\*;

// sound library

Minim minim;

AudioPlayer accSound;

AudioPlayer decSound;

AudioPlayer outSound;

// declare constants

int WINDOW\_WIDTH = 1024;

int WINDOW\_HEIGHT = 768;

// declare images

PImage bgImage;

PImage glider;

PImage glider2;

// declare variables

int xPosition = 1024;

int yPosition = 10;

int objWidth = 90;

int objHeight = 210;

int xDirection = -1;

int yDirection = 1;

int xSpeed = 1;

int ySpeed = 1;

// Background images

//String[] bgImgNames = {"lake.png", "river.png", "bridge.png", "beach.png"};

PImage[] bgImages = new PImage[4];

PImage bgImg1;

PImage bgImg2;

PImage bgImg3;

PImage bgImg4;

// initialization - load images and sound files

void setup()

{

size(WINDOW\_WIDTH, WINDOW\_HEIGHT);

bgImg1 = loadImage("lake.jpg");

bgImg2 = loadImage("river.jpg");

bgImg3 = loadImage("bridge.jpg");

bgImg4 = loadImage("beach.jpg");

bgImages[0] = bgImg1;

bgImages[1] = bgImg2;

bgImages[2] = bgImg3;

bgImages[3] = bgImg4;

bgImage = getRandomImage();

glider = loadImage("glider.png");

glider2 = loadImage("glider2.png");

// noLoop();

// prepare the sound

minim = new Minim(this);

accSound = minim.loadFile("accelerating.mp3");

decSound = minim.loadFile("decreasing.mp3");;

outSound = minim.loadFile("out\_of\_bound.mp3");;

}

// main method to draw the images in the window

void draw()

{

background(bgImage);

xPosition = xPosition + xDirection \* xSpeed;

yPosition = yPosition + yDirection \* ySpeed;

// change the glider image based on its direction

if ( xDirection >= 0 )

{

image(glider2, xPosition, yPosition);

}

else

{

image(glider, xPosition, yPosition);

}

// change direction and the image of the glider when it is "out-of-bound"

if ( (xPosition + objWidth) < 0 )

{

changeForOutOfBound();

xDirection = -xDirection;

image(glider2, xPosition, yPosition);

}

else if ( xPosition > WINDOW\_WIDTH )

{

changeForOutOfBound();

xDirection = -xDirection;

image(glider, xPosition, yPosition);

}

if ( yPosition + objHeight < 0 || yPosition > WINDOW\_HEIGHT )

{

changeForOutOfBound();

yDirection = -yDirection;

}

}

// handle the keystokes

void keyPressed()

{

if (keyCode == LEFT)

{

xDirection = -1;

}

else if (keyCode == RIGHT)

{

xDirection = 1;

}

else if (keyCode == UP)

{

yDirection = -1;

}

else if (keyCode == DOWN)

{

yDirection = 1;

}

else if (keyCode == 'A' || keyCode == 'a')

{

xSpeed++;

ySpeed++;

accSound.rewind();

accSound.play();

}

else if (keyCode == 'S' || keyCode == 's')

{

decSound.rewind();

decSound.play();

if ( xSpeed > 0)

{

xSpeed--;

}

if (ySpeed > 0)

{

ySpeed--;

}

}

}

// change the background image to one of 4 images

PImage getRandomImage()

{

float randNum = random(4);

return bgImages[int(randNum)];

}

// change the background image and play a sound whenever the glider is out of the window

private void changeForOutOfBound()

{

outSound.rewind();

outSound.play();

bgImage = getRandomImage();

}