PONG GAME WITH FLEX SENSOR

Member – Kuiyu ding(62716),

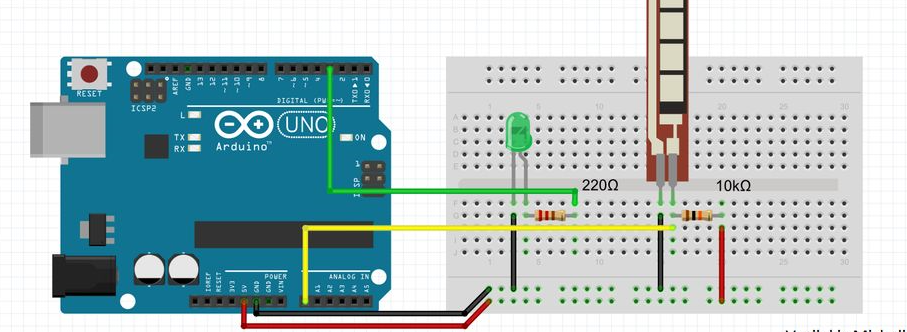
Fahadh Mohamed(62720)

**Flex sensor :**

A flex sensor is generally a flexible strip of conductive ink and material that offers some resistance. This strip is made flexible in order so that it can be bent and alter the resistance. It has two pins P1 and P2.

**Pin connection :**

P1 is connected to the ground pin and P2 is connected to data pin. The data is sent from tm4c123GXL microcontroller to Processing software.



**Processing :**

Processing is afree graphical library and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context.

**Pong game :**

Our aim is to control one paddle with flex sensor based on its variable voltage output , control another paddle by using mouse in processing software. So, we need to give certain conditions with data. Based on this data, we can control the movement of the paddle.

Implementation steps of pong-game :

STEP1 : Draw ball and paddles

STEP2: Move ball and paddles

STEP3: Restrict paddles and bounce off ball

STEP4: Control paddle using sensor parameters

**STEP1 : Draw ball and paddles**：

Using class() syntax, to declare the various parameters of ball and paddle.

1)Set the size and color of ball, paddle

Ball:

class Ball{

void display(){

fill(255,0,0); //fill with (R,G,B), set this color of ball is red

circle(x, y, 50); //center point (x,y), radius equals to 50

}

Paddle:

class Paddle {

float x, y; // position of the paddle

float speedy;

float w = 10;

float h = 150; //define the width and height of paddle

Paddle(float startx, float starty) {

x = startx;

y = starty;

speedy = 0;

}

void display(){

fill(0,255,0); //color of ball is green

rect(x-w/2, y-h/2, w, h); // shape of the paddle is rectangle

}

Background:

void setup(){

size(800,600); // background size 800\*600

}

void draw(){

background(0); //background color is black

}

2) starting point( after centerlized)

Start point of ball and paddles:

void setup(){

ball = new Ball(width/2, height/2);

rq1 = new Paddle(10, height/2);

rq2 = new Paddle(width-10, height/2);

}

the coordinate of ball startpoint: (width/2, height/2);

the coordinate of left paddle startpoint:(5,height/2);

the coordinate of right paddle startpoint:(795,height/2);

**STEP2 : Move Ball and Paddles**

**STEP 3**：**Restrict paddles and bounce off ball**

1. Ball move(Ball bounce):

The ball can move on the x-axis and y-axis, which means it has speed in x, y direction: ball.speedx and ball.speedy.

We want the ball to bounce when it hits the top and bottom edges of the box and when it hits the paddles on the left and right sides, so we need some length calculations.

class Ball{

void movingStep(){

x = x + speedx; // change x position with speed

y = y + speedy;// change y position with speedy

}

X

}

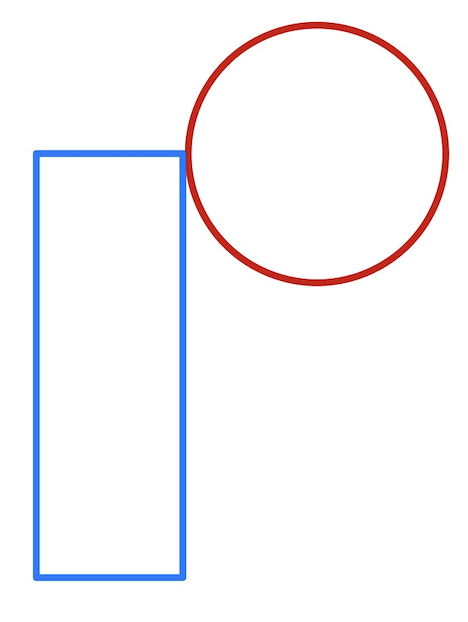
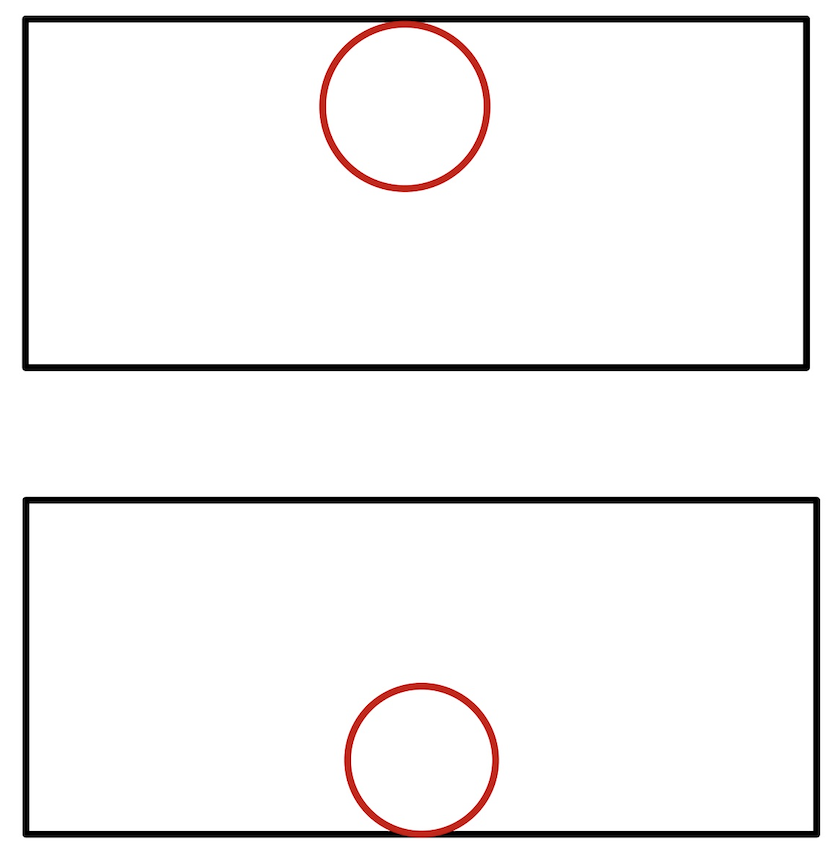
Y

1. ball hit the top/bottom

Ball.y-radius/2=0

Ball.y+radius/2=height

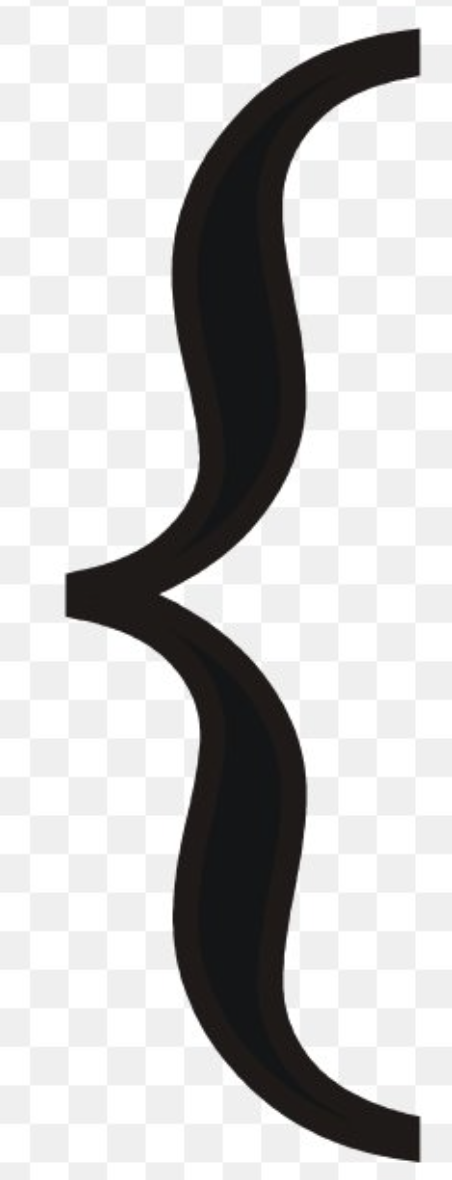
So, the syntax is be like:

if(ball.y + 50/2 > height || ball.y - 50/2 < 0){

ball.speedy = -ball.speedy; // ball bounce back at axis Y

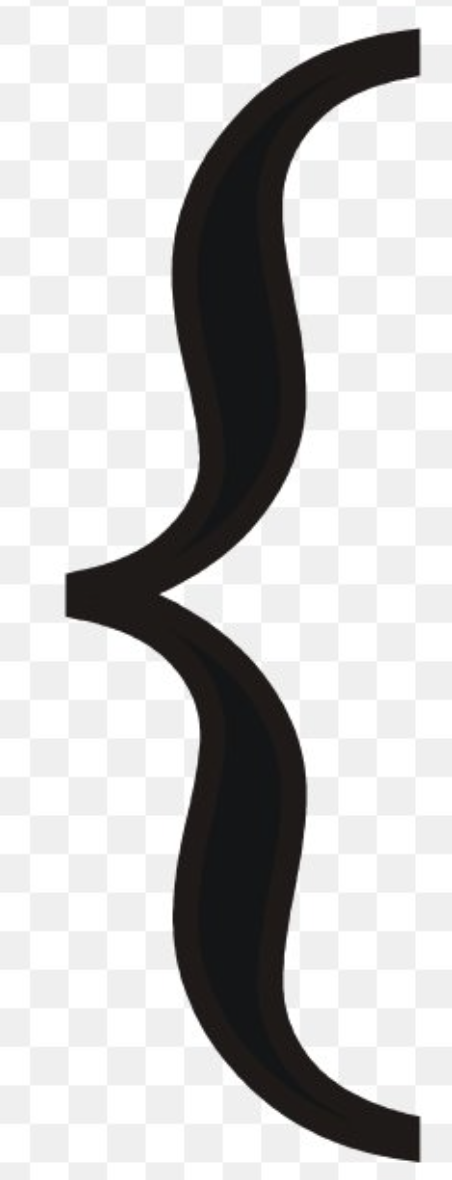
}

b. Ball hit paddles

Ball hit the top edge of left paddle:

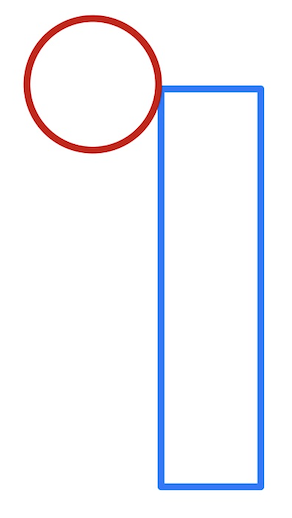
Ball.x-radius/2=width of paddle

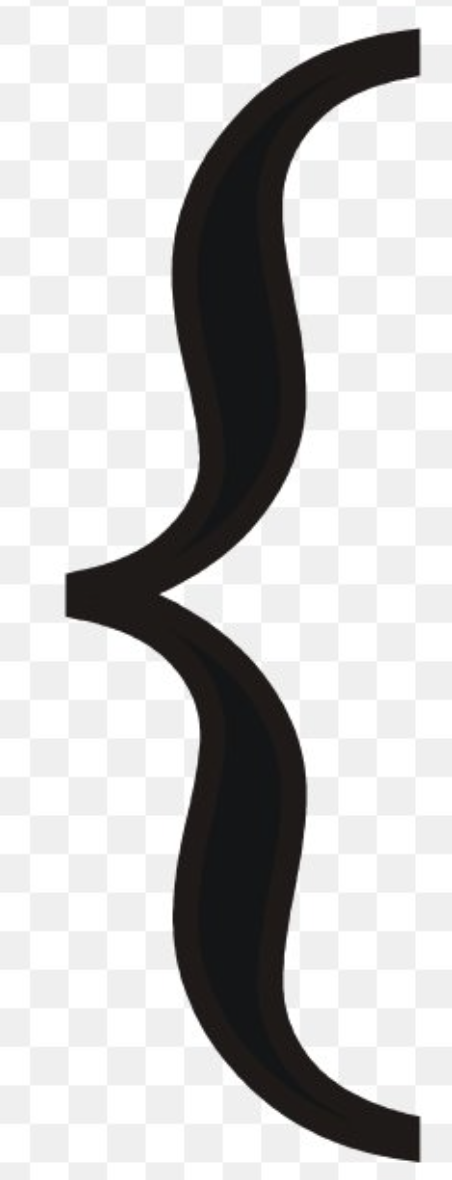
Ball.y=paddley(left)- height of paddle /2

Ball hit the the edge of left paddle:

Ball.x-radius/2=width of paddle

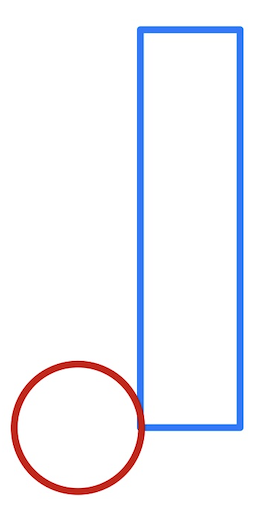
Ball.y=paddley(left)+height of paddle /2

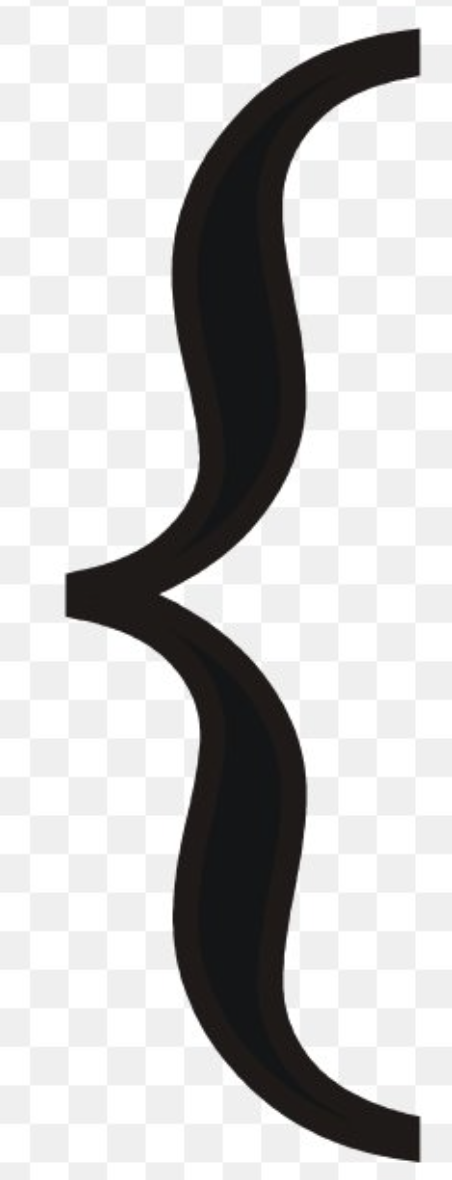


Ball hit the top edge of right paddle:

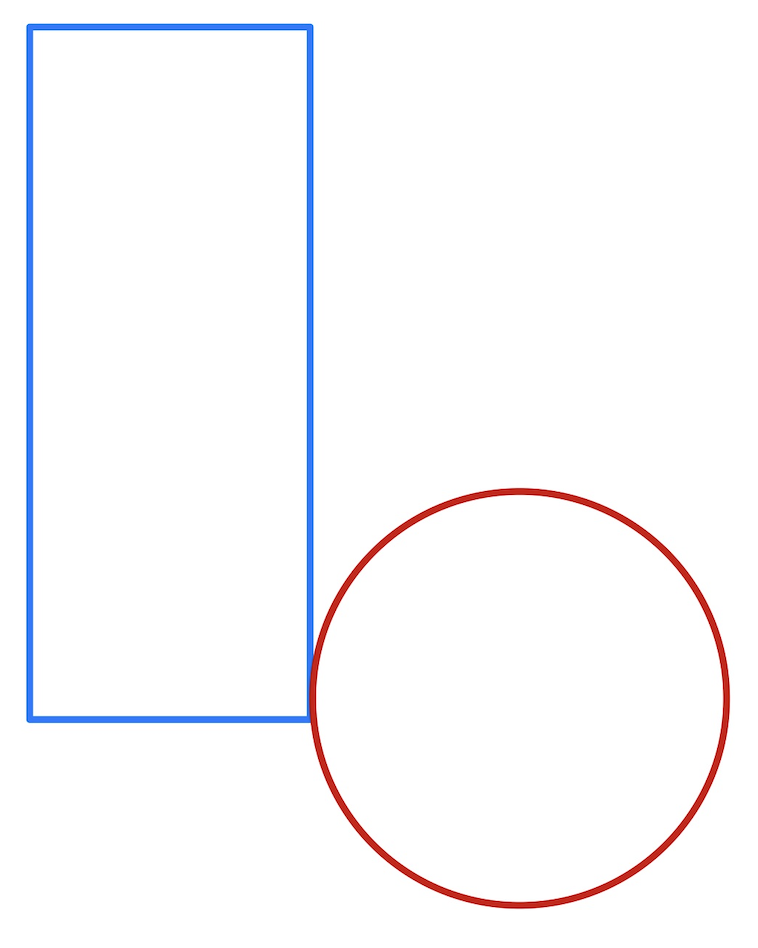
Ball.x+radius/2=paddlex(right)-width of paddle /2

Ball.y=paddley(left)+height of paddle /2



Ball hit the top edge of right paddle:

Ball.x+radius/2=paddlex(right)-width of paddle /2

Ball.y=paddley(left)+height of paddle /2

Abovt all of these, we have syntax:

//control left paddle1

if(ball.y < rq1.y + rq1.h/2){//bottom paddle

if(ball.y > rq1.y - rq1.h/2){//top paddle

if(ball.x-50/2 < rq1.x+rq1.w/2){

ball.speedx = -ball.speedx;

ball.speedy = map(ball.y - rq1.y, -rq1.h/2, rq1.h/2, -15, 15);

}

// control right paddle2

if(ball.y < rq2.y + rq2.h/2){

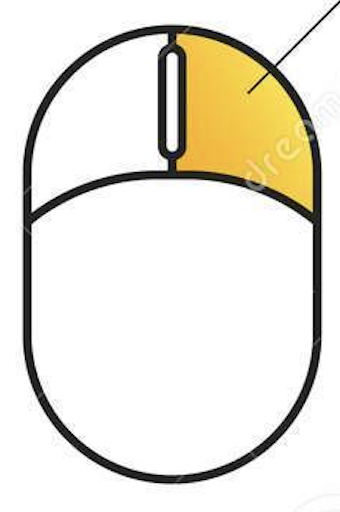
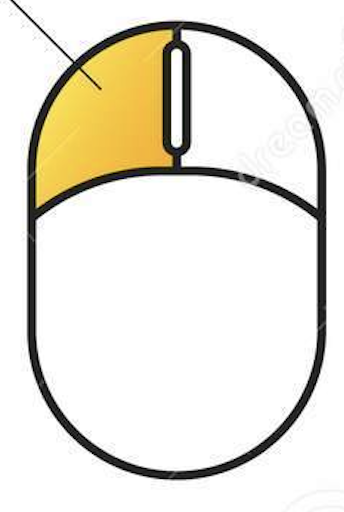
if(ball.y > rq2.y - rq2.h/2){

if(ball.x+50/2 > rq2.x-rq2.w/2){

ball.speedx = - ball.speedx;

ball.speedy = map(ball.y - rq2.y, -rq2.h/2, rq2.h/2, -15, 15);

}

// after ball hit the paddle, ball.speedy is not constant, It is proportional to the difference between the y coordinate of the center of the ball and the y coordinate of the center of the paddle, which also means that if we catch the ball with the edge of the paddle, the speed of the ball will be faster than if we catch the ball with the center of the paddle. This adds more interest to the game, as catching the ball from the edge is a high-risk, high-reward affair.

mouseButton==left, speedY>0

mouseButton==right, speedY<0

// Amplify velocity from (-rq.h/2, rq.h/2) to (-15, 15) using map syntax

2) Paddle move(paddle strict)

Simply Use mousePressed and mouseReleased syntaxto control the paddle on right side.

void mousePressed(){

if(mouseButton==LEFT){

rq2.speedy = 5; //when press left mouse , paddle move toward bottom

}

if (mouseButton==RIGHT){

rq2.speedy = -5; //when press right mouse , paddle move toward top

}

}

void mouseReleased(){

if(mouseButton==LEFT){

rq2.speedy = 0;//when release left mouse , paddle stop moveing toward bottom

}

if (mouseButton==RIGHT){

rq2.speedy = 0; //when release right mouse , paddle stop moveing toward top

}

}

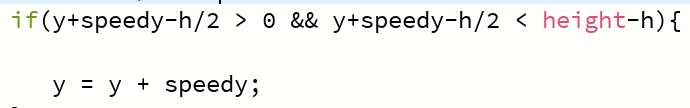
Restrict paddles:

Control the movement of the paddle in the y-axis direction not to exceed the 800\*600 game area.

Only When Paddley - height of paddle/2 >0 and paddley + height of paddle /2<0, paddley = paddley +speed.

Ball drop:

When ball hit the left/right side of background, tha ball will drop , and restart from centeral point of background.

And score 1 point to the winner.

if(ball.x + 50/2 > width){

ball.x = width/2;

ball.y = height/2;

scoreleft = scoreleft + 1;

}

if(ball.x - 50/2 < 0){

ball.x = width/2;

ball.y = height/2;

scoreright = scoreright +1 ;

}

**STEP4 : Control paddle Using sensor data**

1. load a serial library into a Processing sketch for reading serial data.

import processing.serial.\*;

2)using syntax :  
Serial(parent, portName, baudRate)

|  |  |
| --- | --- |
| **parent** | typically use "this" |
| **baudRate** | 9600 is the default |
| **portName** | name of the port (COM1 is the default |

Implementation:

serialPort = new Serial(this, Serial.list()[0], 9600);

buffer.until(inByte)

readStringUntil()

Sets a specific byte to buffer until before calling serialEvent() and read data from buffer.

serialPort.bufferUntil ( '\n' );

void serialEvent(Serial serialPort){

float actionSer = float(serialPort.readStringUntil(‘\n')); // read data

println(actionSer);

if(actionSer < 400){

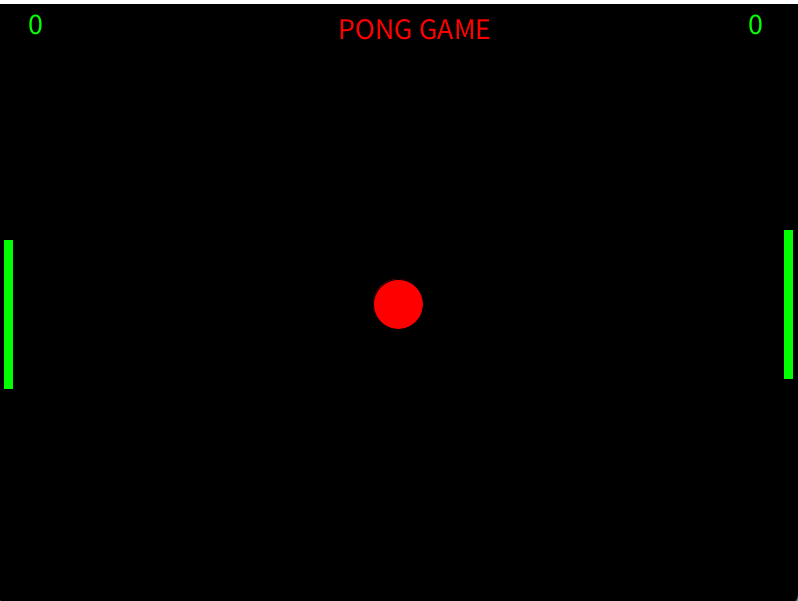
rq1.speedy = 5; // if flex sensor turn toward one side , paddle left move down;

}

if(actionSer > 400){

rq1.speedy = -5; // if flex sensor turn toward another side , paddle left move up;

}



**Internet feature: Twitter**

Twitter is an social media, which user can share their thoughts. In this pong game, player 1 is played with flex sensor and player two is played with mouse. If player 1 won the game, the tweet will be posted in twitter automatically. We can post this tweet via twitter API.

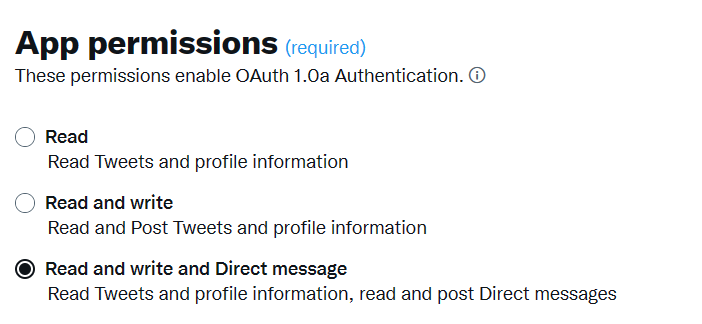
**API(Application Programming Interface) :**

API is a software interface that allows two applications to interact with each other without any user intervention. In simple terms, API means a software code that can be accessed or executed. API is defined as a code that helps two different software’s to communicate and exchange data with each other.

**How do we get API key ?**

Initially, we have to create an twitter account to post the results. Then, we have to log in twitter developer account and apply for API keys. There is three types of access in twitter developer mode. There are essential mode, elevated mode, and academic research mode. By default, we will get access for essential mode. But, To post the result from processing software, we need elevated mode access.

We have to apply for elevated mode access, it will take some days to approve our request. Once, our request is approved we can directly post our results via API keys. It is very important to keep the key secretly, if someone got the keys they will have access to our twitter account. We can always regenerate new keys if we lost our keys. We also need to change the app permissions to read, write and direct message to tweet in twitter.



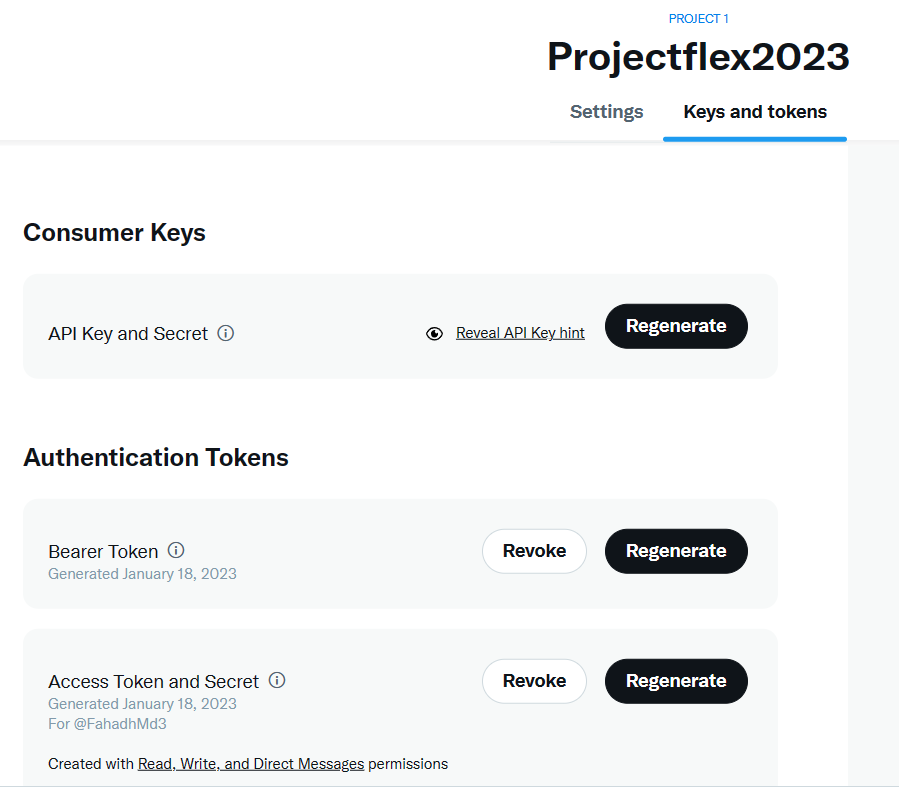
There are four types of keys, we are using to post the tweet. There are consumer key, consumer secret key, access token key, access token secret key.

**Consumer Key and Consumer Secret Key:**

The API Key and Secret (also known as Consumer Key and Secret) are the most fundamental credentials required to access the Twitter API. These credentials act as the username and password for your Twitter App.

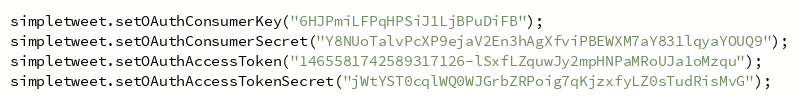
**Access Token and Access Token Secret Key:**

The access token represents the authorization of a specific application to access specific parts of a user’s data.



**Simpletweet :**

To post the tweet in twitter, we use simpletweet library in processing software. First, we need to include our keys in processing software buy using simpletweet function.



Now, we need to write programming for post the tweet. The program should be, if player 1 scored == 10 points, the tweet will be posted.



Above program shows, if scoreleft == 10, the tweet will be posted as “Player 1 won against Player 2”. Same as, if scoreright == 10, the tweet will be posted as “Player 2 won against Player 1”.



**Result :**

After winning the game, the tweet is automatically posted in twitter account via API.

