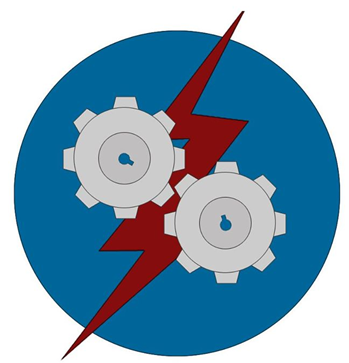
**E.T N°32 D. E°14**

**Gral. José de San Martín**

Proyecto Informático

Carpeta de campo

**Juego Arcade**



**Profesor:** Gonzalo Consorti

**Integrantes:** Francisco Rabito, Alejandro Ferreira y Lucas Bravo

**Año y División:** 4° 1

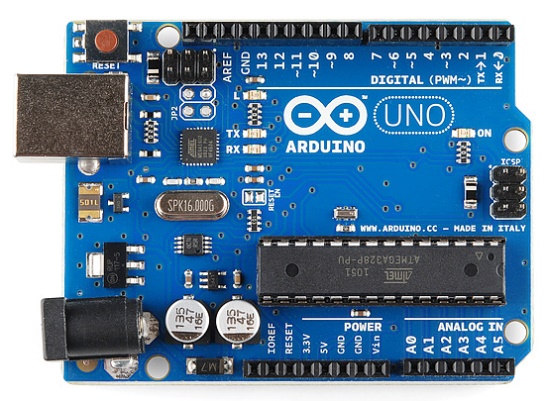
**Objetivo del proyecto:** La finalidad del proyecto es la de programar un mini juego arcade con arduino y processing, utilizando los materiales adquiridos en clase, conectándolos con el arduino y probando diferentes códigos.

Empezamos investigando sobre el proyecto en sí. Que juego podríamos programar, que componentes tendríamos que utilizar, tratando de hacer un juego simple.  
Buscamos videos sobre distintos tipos de juegos programados en arduino y luego llegamos a la conclusión de que el proyecto duraría 2 trimestres debido a que el juego que íbamos a programar tendría que ser hecho en processing.  
El juego era el “Snake” ya que la parte de la programación no tiene mucha complejidad, además de ser un juego simple, pero con mecánicas interesantes.  
Decidimos utilizar 4 botones que controlarán a la serpiente en vez de un joystick, debido a la dificultad que tendría este al programarse y por conveniencia sobre los controles del juego.

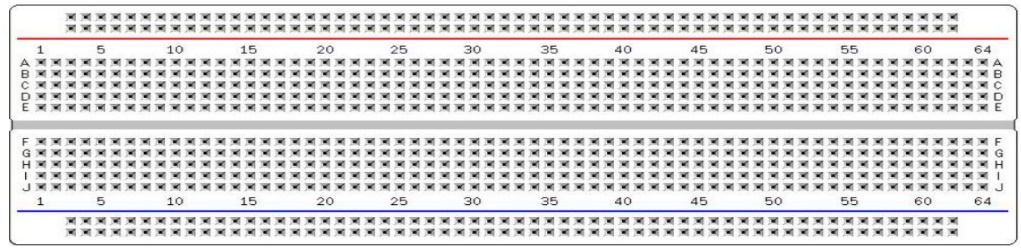
**Descripción del juego:** El jugador controla a una [serpiente](https://es.wikipedia.org/wiki/Serpiente), que recorre un campo limitado, recogiendo manzanas o algún otro elemento, tratando de evitar golpear a su propia cola o las paredes que rodean el área de juego. Cada vez que la serpiente se come un pedazo de manzana, la cola crece más, provocando que aumente la dificultad del juego.

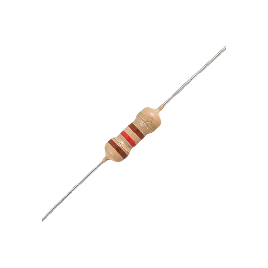
**Páginas visitadas durante la realización del proyecto:**   
https://processing.org/  
https://www.tinkercad.com/  
https://www.youtube.com/watch?v=AaGK-fj-BAM  
 **Componentes- Imágenes-Código**

En las clases de proyecto informático estuvimos probando varios códigos de arduino y processing, además de utilizar los materiales necesarios para el proyecto los cuales son:

\_Un Arduino UNO

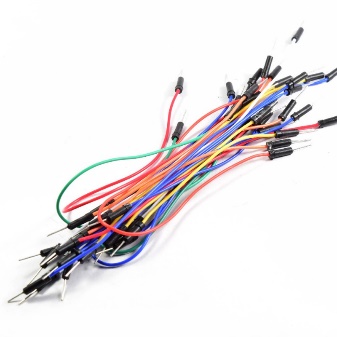
\_1 Protoboard



\_Un led \_4 Botones \_1 Resistencia





\_Cables \_Un cable USB

Terminamos el código de arduino que planeamos utilizar de testeo, y conectamos los 4 botones y un led con el arduino y un protoboard para corroborar la funcionalidad utilizando el siguiente código:

void setup(){

pinMode(12, INPUT); //boton1

pinMode(11, INPUT); //boton2

pinMode(10, INPUT); //boton3

pinMode(9, INPUT); //boton4

pinMode(4, OUTPUT); //led

}

void loop(){

if(digitalRead(12) == LOW){

digitalWrite(4,HIGH);

delay(100);

}

else if(digitalRead(11) == LOW){

digitalWrite(4,HIGH);

delay(100);

}

else if(digitalRead(10) == LOW){

digitalWrite(4,HIGH);

delay(100);

}

else if(digitalRead(9) == LOW){

digitalWrite(4,HIGH);

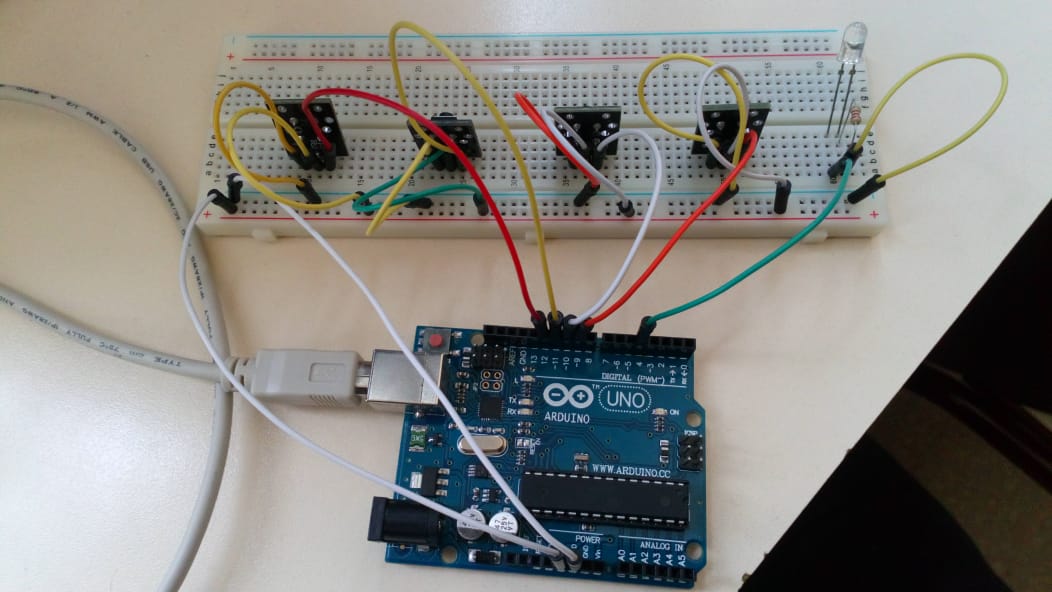
delay(100);

}

digitalWrite(4,LOW);

}

Y conectándolo de esta manera:



También aprendimos más sobre la función “Serial” la cual permite conectar al arduino con el processing.

Esta fue una prueba que realizamos para comprobar el funcionamiento de los componentes y como trabajaba el arduino. Pero luego terminamos por hacer el código el cual se comunicaría con el processing a través de la función Serial:

**CODIGO FINAL DE ARDUINO:**

int x=0;

void setup(){

Serial.begin(9800);

pinMode(12, INPUT); //boton1

pinMode(11, INPUT); //boton2

pinMode(9, INPUT); //boton4

pinMode(10, INPUT); //boton3

}

void loop(){

if(digitalRead(12) == LOW){

x = 1;

delay(10);

}

else if(digitalRead(11) == LOW){

x = 2;

delay(10);

}

else if(digitalRead(10) == LOW){

x = 3;

delay(10);

}

else if(digitalRead(9) == LOW){

x = 4;

delay(10);

}

else {

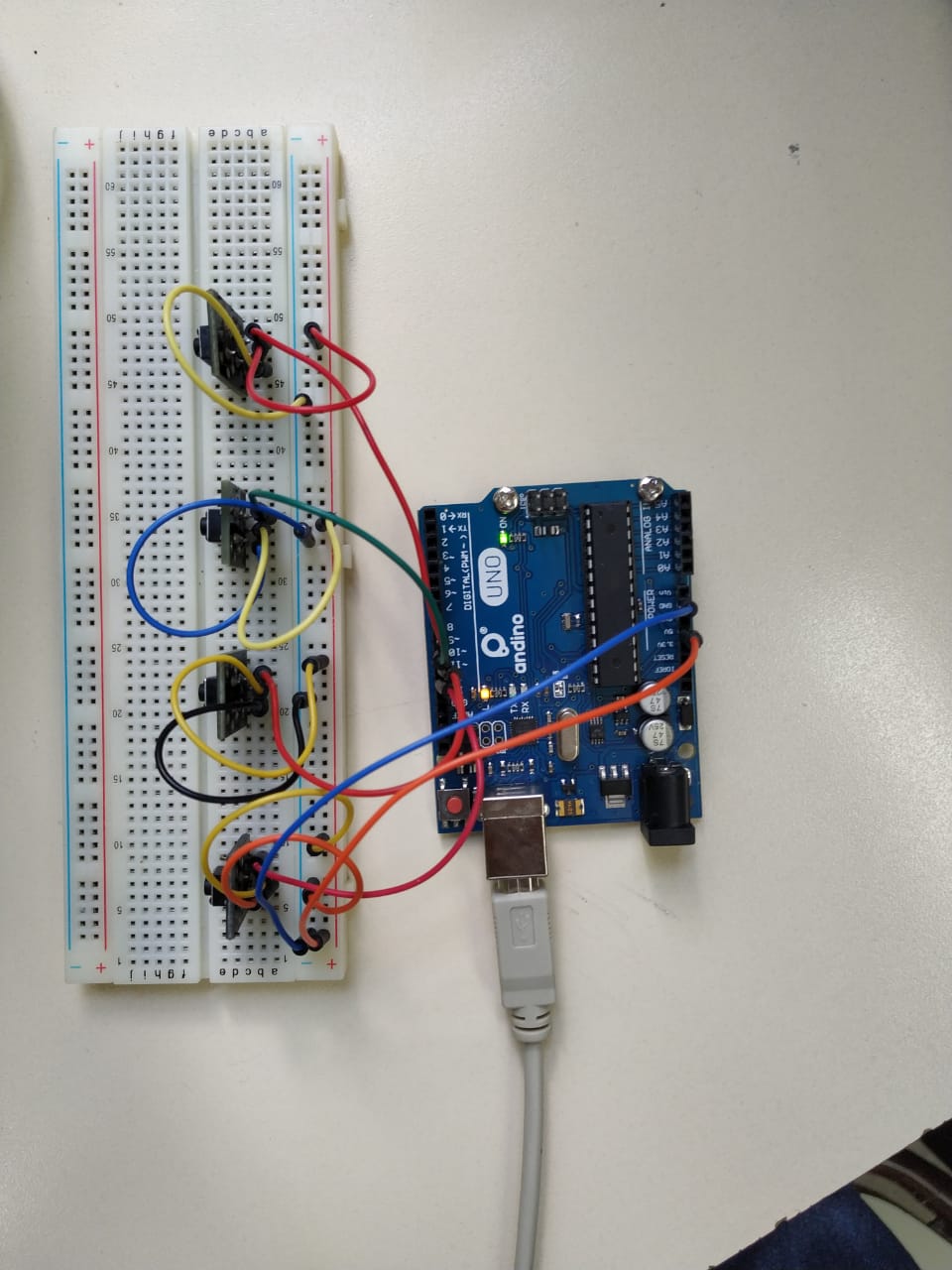
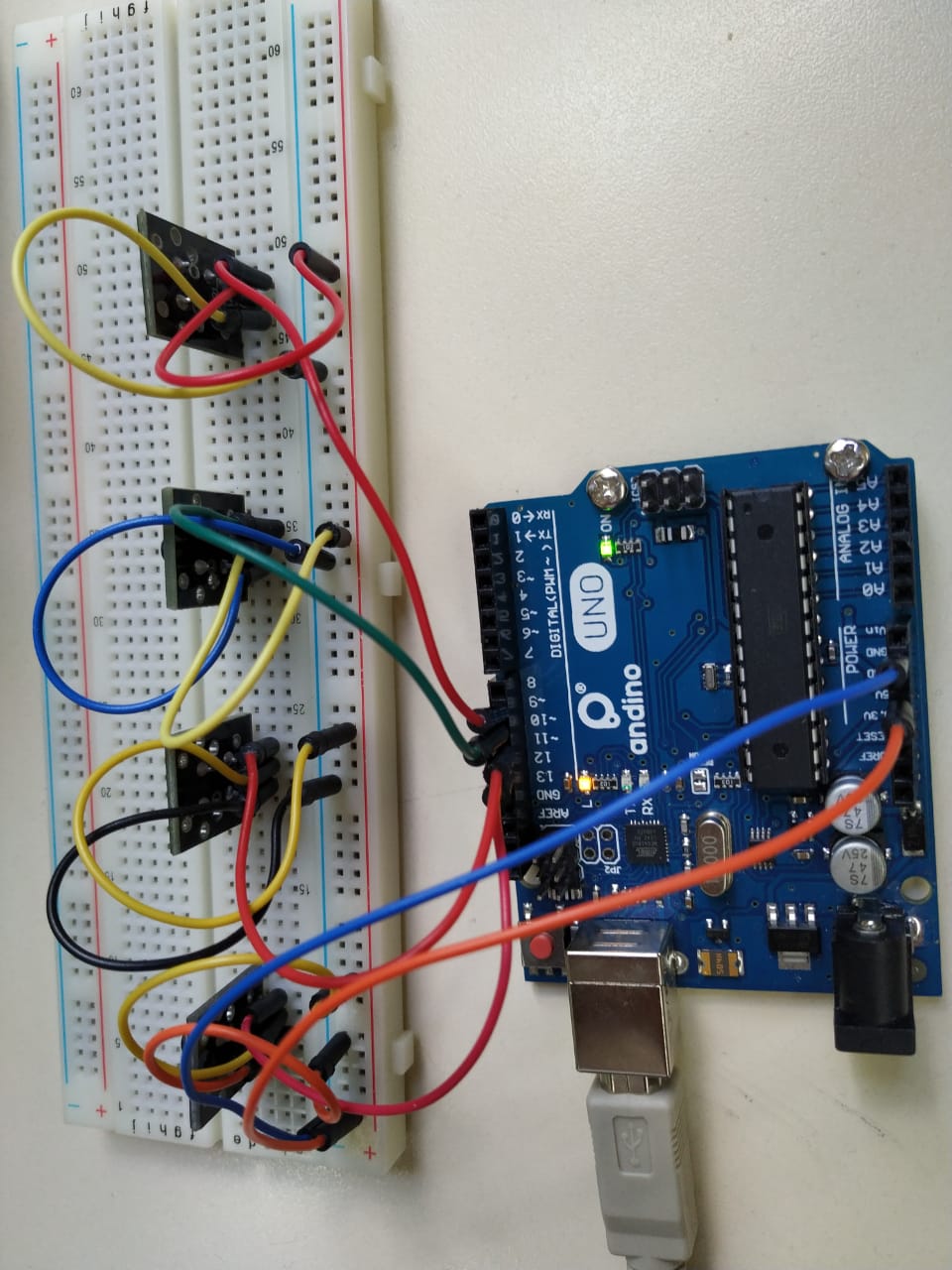
x = 0;

delay(10);

}

Serial.print(x); Serial.print('.');

}

Y lo conectamos de la siguiente forma: 

Luego de finalizar con la parte del Arduino y el led, nos centramos más en el código del juego en processing y estuvimos viendo varias de las funciones del lenguaje.

Ya después de aprender las funciones necesarias para crear el código en processing, proseguimos a crear el snake en dicho programa y probarlo, utilizando el siguiente código:

**CODIGO FINAL DE PROCESSING CONTROLADO A TRAVES DEL TECLADO:**

**//Clase sneke:**

public class sneke {

int[] x = new int[900];

int[] y = new int[900];

public sneke(){

for(int i = 1 ; i < 900 ; i++){

x[i] = -1;

y[i] = -1;

}

x[0] = 15;

y[0] = 15;

}

public int[] getX(){

return this.x;

}

public int[] getY(){

return this.y;

}

public void setX(int[] xP){

this.x = xP;

}

public void setY(int[] yP){

this.y = yP;

}

}

**//Codigo principal del juego:**

import java.io.BufferedInputStream;

import java.io.FileNotFoundException;

sneke snake = new sneke(); //objeto snake

final int sum = 1; //la cantidad de cuadrados que se suman al comer una manzana

int total=1; //almacena el puntaje total del jugador

int highscore=0; //almacena el highscore temporal

int[] auxX; //auxiliar del vectorX en el objeto snake

int[] auxY; //auxiliar del vectorY en el objeto snake

int manzanax = (int)random(0,29); //posicion de la manzana en x

int manzanay = (int)random(0,29); //posicion de la manzana en y

int tecla = -1; //lee la tecla presionada

int tecla2 = -1; //auxiliar de tecla

boolean muerto = false; //controla si el jugador esta vivo o muerto

boolean youWin = false; //controla si el jugador llego al puntaje maximo

int w=30,h=30; //width(ancho) y height(altura) de la pantalla

int bs=20; //block size(tamaño en pixeles de cada unidad en el juego)

//--------------------------------------------------------------------------------------------------------------------------------------------

void setup(){

size(600, 600);

frameRate(10);

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void draw(){

if(total>=899){

youWin = true;

muerto = true;

}

if(muerto==true) gameover();

auxY = snake.getY(); //Lee los datos del objeto snake en el vectorY y los almacena en una variable auxiliar

auxX = snake.getX(); //Lee los datos del objeto snake en el vectorX y los almacena en una variable auxiliar

snakeCollision(); //controla que el snake no se choque con el mismo

screenLimits(); //limites de pantalla (si el snake llega al limite pierde)

if(muerto==false){

background(0);

showScore(); //muestra el puntaje del jugador en tiempo real

//modificacion de los datos

//posiciona todas las partes del snake

for(int i = total-1 ; i > 0 ; i--){

auxY[i] = auxY[i-1];

auxX[i] = auxX[i-1];

}

snakeDir(); //lee la tecla presionada y selecciona la direccion en la cual se movera el snake

//dibuja el snake

for(int i = 0 ; i < total ; i++){

rect(auxX[i]\*bs, auxY[i]\*bs, bs, bs);

}

//cuando el snake come una manzana se genera una nueva posicion para la manzana

if(auxX[0] == manzanax && auxY[0] == manzanay){

pickNewLocation(); //elige una nueva posicion para la manzana

total += sum; //a total se le suma el valor de la variable sum(descripcion arriba)

}

manzana(); //dibuja la manzana

}

snake.setY(auxY); //agrega los datos modificados en el vectorY al objeto snake

snake.setX(auxX); //agrega los datos modificados en el vectorX al objeto snake

}

//--------------------------------------------------------------------------------------------------------------------------------------------

//lee el codigo de la tecla presionada y lo almacena en una variable

void keyPressed(){

tecla = keyCode;

if(tecla==38 && auxY[0]-1!=auxY[1] || tecla==40 && auxY[0]+1!=auxY[1] || tecla==39 && auxX[0]+1!=auxX[1] || tecla==37 && auxX[0]-1!=auxX[1]){

tecla2=tecla; //controla que la tecla presionada sea una de las flechas y que el snake no se mueva en reversa

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void snakeDir(){

if(muerto==false){

switch(tecla2){

case 38:

auxY[0]--;

break;

case 40:

auxY[0]++;

break;

case 39:

auxX[0]++;

break;

case 37:

auxX[0]--;

break;

default:

break;

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void pickNewLocation(){

boolean falla=true;

while(falla==true){

manzanax = (int)random(0,29);

manzanay = (int)random(0,29);

falla = false;

for(int i = 0; i < total; i++){

if(manzanax==auxX[i] && manzanay==auxY[i]){

falla=true;

}

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void manzana(){

fill(255,0,0);

rect(manzanax\*bs,manzanay\*bs,bs,bs);

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void screenLimits(){

if (auxX[0]>=w || auxX[0]<=-1 || auxY[0]>=h || auxY[0]<=-1){

muerto = true;

gameover();

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void snakeCollision(){

for(int i = 1; i < total; i++){

if(auxX[0] == auxX[i] && auxY[0] == auxY[i]){

muerto = true;

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void showScore(){

fill(255);

textSize(15);

text("SCORE: "+ total,10,20);

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void highscore(){

if(total>highscore){

highscore=total;

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

//cuando el snake muere/gana se muestra una pantalla de game over/you win y se setean todas las variables como al inicio

void gameover(){

highscore();

background(0);

fill(255);

textSize(50);

if(youWin==false){

text("GAME OVER",155,200);

}else{

text("YOU WIN",190,200);

}

textSize(30);

text("SCORE: "+total,218,300);

text("HIGHSCORE: "+ highscore,185,335);

textSize(20);

text("PRESS SPACE TO RESTART",170,400);

for(int i = 1 ; i < total ; i++){

auxX[i] = -1;

auxY[i] = -1;

}

if(tecla == 32){

auxX[0] = 15;

auxY[0] = 15;

muerto = false;

youWin = false;

total = 1;

tecla = -1;

tecla2 = -1;

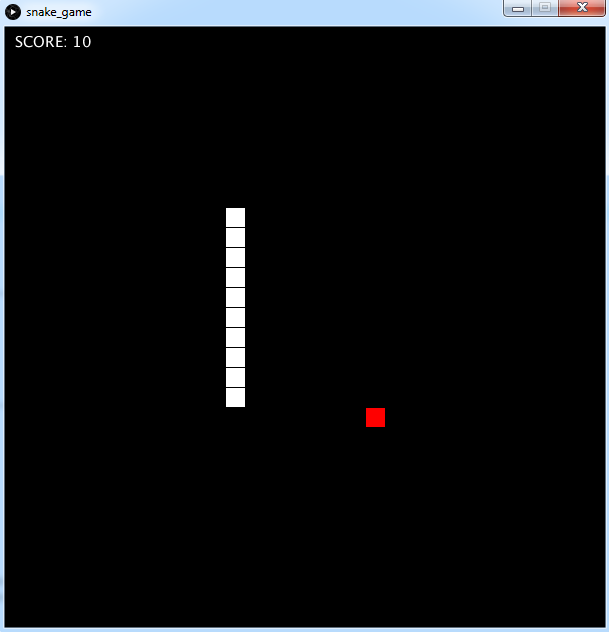
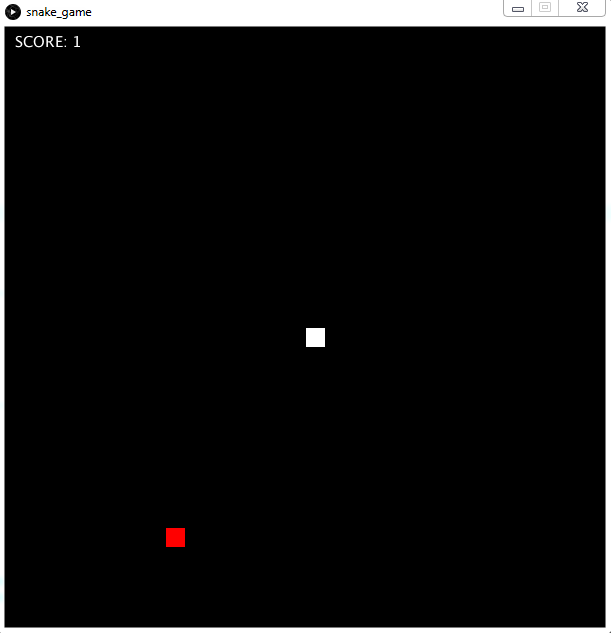
manzanax = (int)random(0,29);

manzanay = (int)random(1,29);

}

}

Capturas del juego:



Una vez que terminamos de programar el juego, pasamos a adaptarlo para que funcione con el código del arduino:

**CODIGO FINAL DE PROCESSING CONTROLADO A TRAVES DEL ARDUINO:**

**//Clase sneke:**

public class sneke {

int[] x = new int[900];

int[] y = new int[900];

public sneke(){

for(int i = 1 ; i < 900 ; i++){

x[i] = -1;

y[i] = -1;

}

x[0] = 15;

y[0] = 15;

}

public int[] getX(){

return this.x;

}

public int[] getY(){

return this.y;

}

public void setX(int[] xP){

this.x = xP;

}

public void setY(int[] yP){

this.y = yP;

}

}

**//Codigo principal del juego**

import java.io.BufferedInputStream;

import java.io.FileNotFoundException;

import processing.serial.\*; //importa la libreria serial

Serial puerto; //puerto serial

sneke snake = new sneke(); //objeto snake

final int sum = 1; //la cantidad de cuadrados que se suman al comer una manzana

int total=1; //almacena el puntaje total del jugador

int highscore=0; //almacena el highscore temporal

int[] auxX; //auxiliar del vectorX en el objeto snake

int[] auxY; //auxiliar del vectorY en el objeto snake

int manzanax = (int)random(0,29); //posicion de la manzana en x

int manzanay = (int)random(0,29); //posicion de la manzana en y

int tecla = -1;

int dato2 = 0; //auxiliar del dato leido por el puerto serial

boolean muerto = false; //controla si el jugador esta vivo o muerto

boolean youWin = false; //controla si el jugador llego al puntaje maximo

int w=30,h=30; //width(ancho) y height(altura) de la pantalla

int bs=20; //block size(tamaño en pixeles de cada unidad en el juego)

//--------------------------------------------------------------------------------------------------------------------------------------------

void setup(){

puerto= new Serial(this, "COM3", 9800); //se crea un puerto serial

puerto.bufferUntil('.'); //genera un evento serial hasta que se identifica

size(600, 600);

frameRate(10);

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void draw(){

if(total>=899){

youWin = true;

muerto = true;

}

if(muerto==true) gameover();

auxY = snake.getY(); //Lee los datos del objeto snake en el vectorY y los almacena en una variable auxiliar

auxX = snake.getX(); //Lee los datos del objeto snake en el vectorX y los almacena en una variable auxiliar

snakeCollision(); //controla que el snake no se choque con el mismo

screenLimits(); //limites de pantalla (si el snake llega al limite pierde)

if(muerto==false){

background(0);

showScore(); //muestra el puntaje del jugador en tiempo real

//modificacion de los datos

//posiciona todas las partes del snake

for(int i = total-1 ; i > 0 ; i--){

auxY[i] = auxY[i-1];

auxX[i] = auxX[i-1];

}

snakeDir(); //lee la tecla presionada y selecciona la direccion en la cual se movera el snake

//dibuja el snake

for(int i = 0 ; i < total ; i++){

rect(auxX[i]\*bs, auxY[i]\*bs, bs, bs);

}

//cuando el snake come una manzana se genera una nueva posicion para la manzana

if(auxX[0] == manzanax && auxY[0] == manzanay){

pickNewLocation(); //elige una nueva posicion para la manzana

total += sum; //a total se le suma el valor de la variable sum(descripcion arriba)

}

manzana(); //dibuja la manzana

}

snake.setY(auxY); //agrega los datos modificados en el vectorY al objeto snake

snake.setX(auxX); //agrega los datos modificados en el vectorX al objeto snake

}

//--------------------------------------------------------------------------------------------------------------------------------------------

//lee el codigo de la tecla presionada y lo almacena en una variable

void keyPressed(){

tecla = keyCode;

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void serialEvent(Serial puerto) {

String datoIN= puerto.readStringUntil('.');

datoIN = trim( datoIN.substring(0,1) );

int dato = Integer.parseInt(datoIN);

if(dato==4 && auxY[0]-1 != auxY[1] || dato==2 && auxY[0]+1!=auxY[1] || dato==1 && auxX[0]+1!=auxX[1] || dato==3 && auxX[0]-1!=auxX[1]){

dato2=dato; //controla que el snake no se mueva en reversa

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void snakeDir(){

if(muerto==false){

switch(dato2){

case 4:

auxY[0]--;

break;

case 2:

auxY[0]++;

break;

case 1:

auxX[0]++;

break;

case 3:

auxX[0]--;

break;

default:

break;

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void pickNewLocation(){

boolean falla=true;

while(falla==true){

manzanax = (int)random(0,29);

manzanay = (int)random(0,29);

falla = false;

for(int i = 0; i < total; i++){

if(manzanax==auxX[i] && manzanay==auxY[i]){

falla=true;

}

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void manzana(){

fill(255,0,0);

rect(manzanax\*bs,manzanay\*bs,bs,bs);

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void screenLimits(){

if (auxX[0]>=w || auxX[0]<=-1 || auxY[0]>=h || auxY[0]<=-1){

muerto = true;

gameover();

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void snakeCollision(){

for(int i = 1; i < total; i++){

if(auxX[0] == auxX[i] && auxY[0] == auxY[i]){

muerto = true;

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void showScore(){

fill(255);

textSize(15);

text("SCORE: "+ total,10,20);

}

//--------------------------------------------------------------------------------------------------------------------------------------------

void highscore(){

if(total>highscore){

highscore=total;

}

}

//--------------------------------------------------------------------------------------------------------------------------------------------

//cuando el snake muere/gana se muestra una pantalla de game over/you win y se setean todas las variables como al inicio

void gameover(){

highscore();

background(0);

fill(255);

textSize(50);

if(youWin==false){

text("GAME OVER",155,200);

}else{

text("YOU WIN",190,200);

}

textSize(30);

text("SCORE: "+total,218,300);

text("HIGHSCORE: "+ highscore,185,335);

textSize(20);

text("PRESS SPACE TO RESTART",170,400);

for(int i = 1 ; i < total ; i++){

auxX[i] = -1;

auxY[i] = -1;

}

if(tecla == 32){

auxX[0] = 15;

auxY[0] = 15;

muerto = false;

youWin = false;

total = 1;

tecla = -1;

manzanax = (int)random(0,29);

manzanay = (int)random(1,29);

}

}