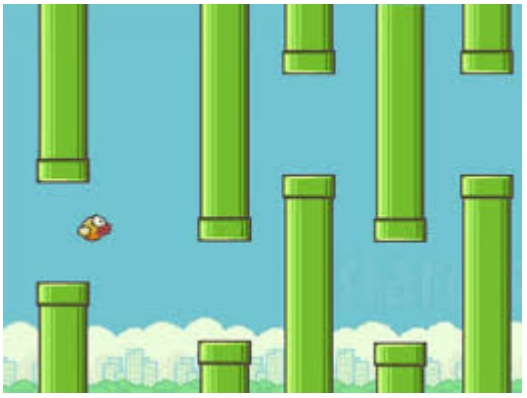
Ejercicio 1: Juego Flappy Bird

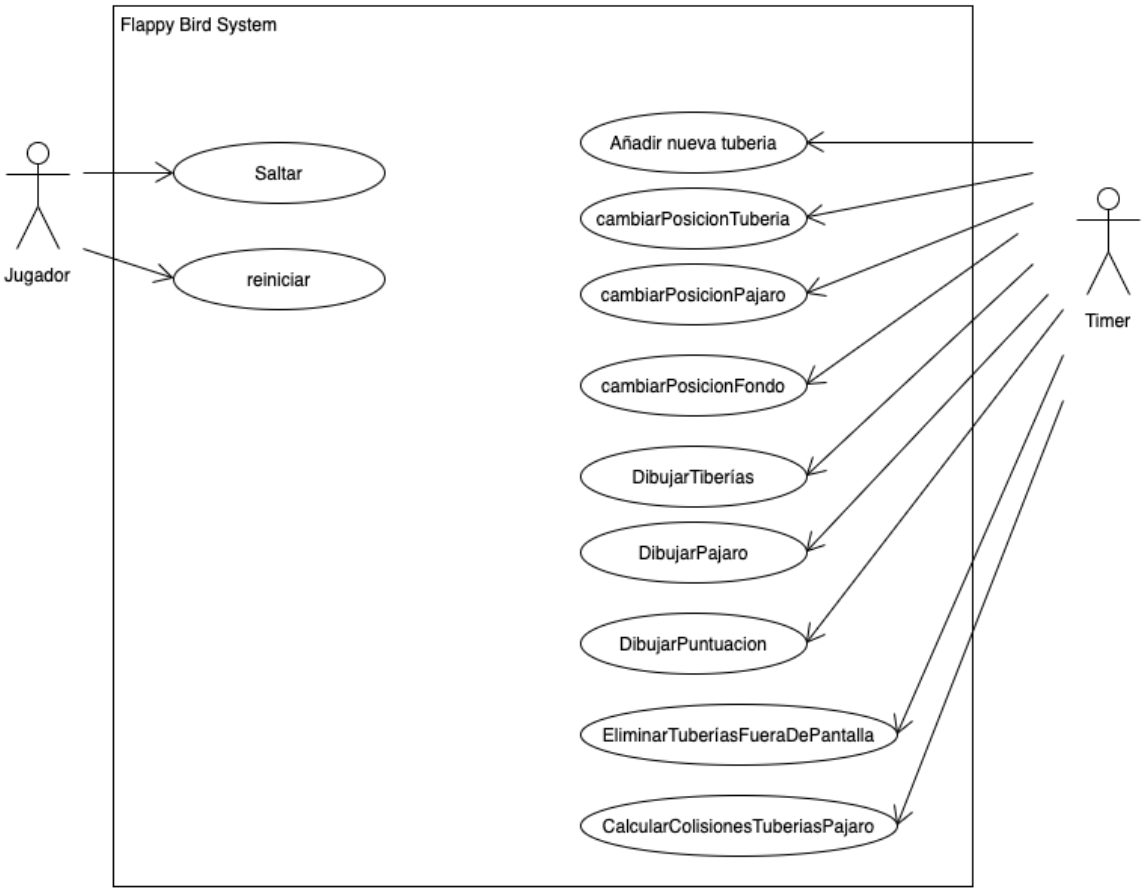


Puedes ver los ficheros en la carpeta compartida:

Entornos de desarrollo-->Juegos clasicos-->FlappyBird

Parte 1: Casos de uso

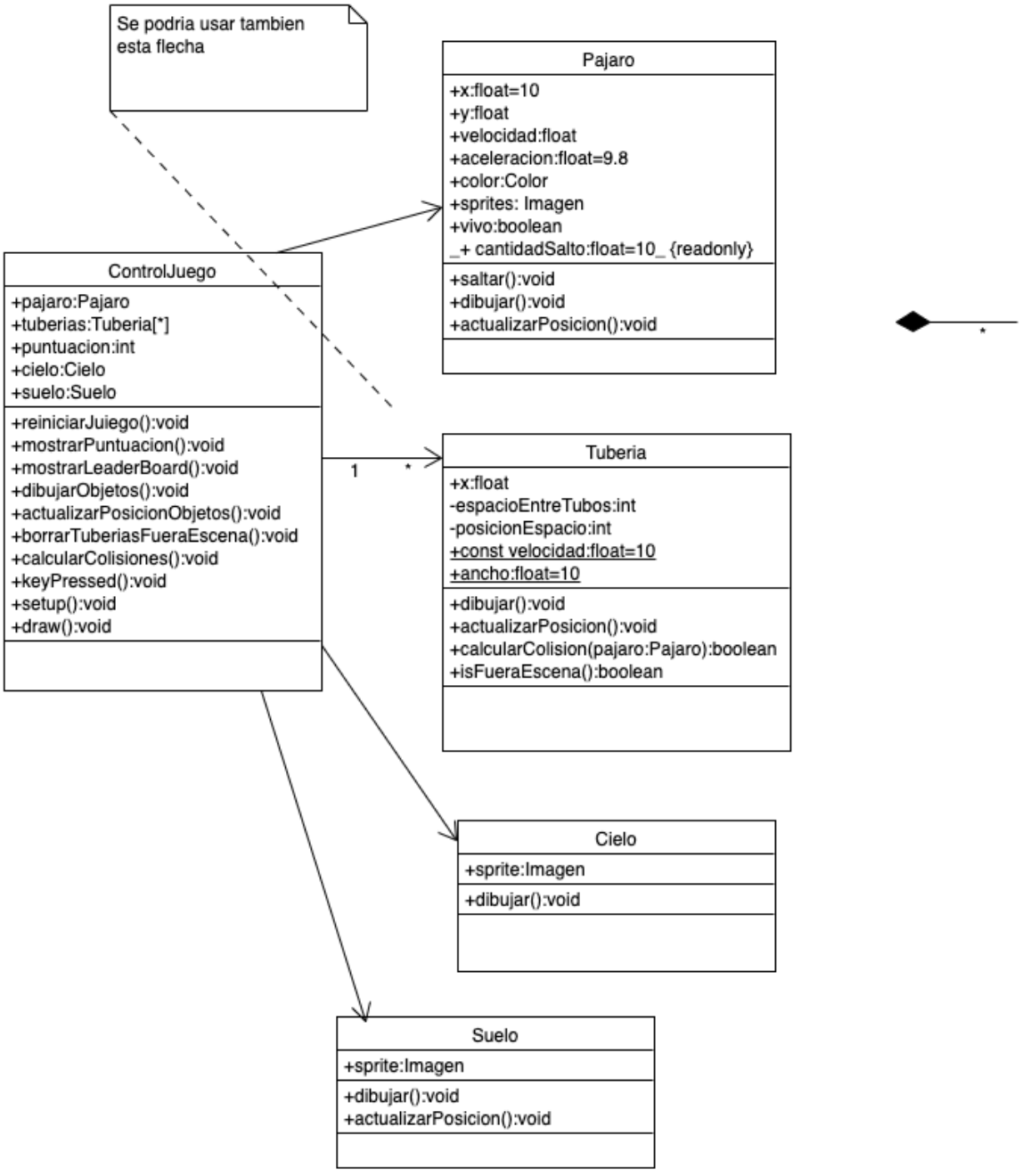
Realiza el diagrama de casos de uso del juego Flappy Bird.
Solución:



Parte 2: Diagrama clases

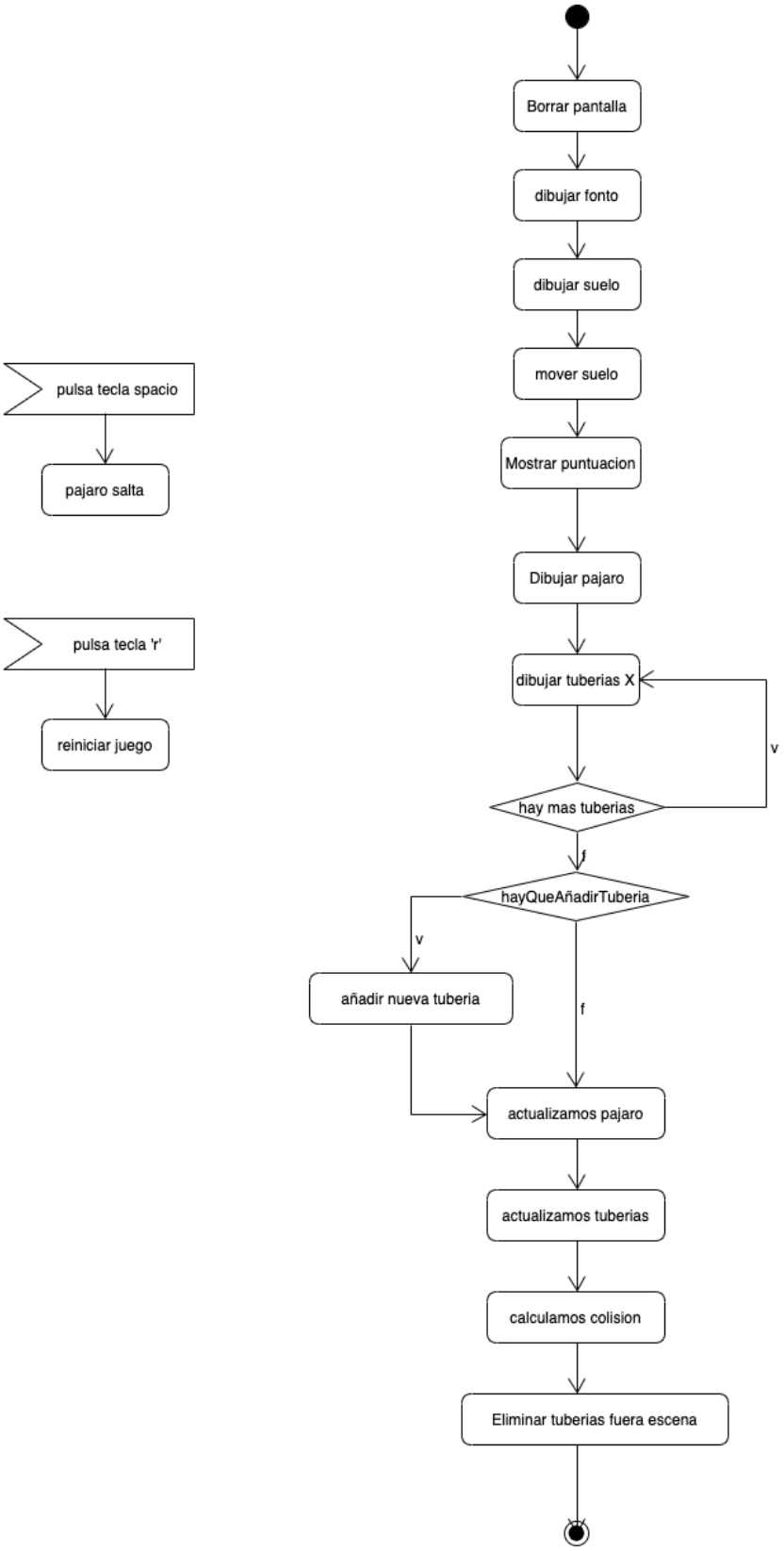
Construye el diagrama de clases UML

Solución:



Parte 3: Diagrama de actividad

Construye el diagrama uml de actividad (solamente del programa principal). Esto se corresponde con la actividad que se realiza en 1 fotograma


Borrar pantalla
dibujar fonto
dibujar suelo
mover suelo
Mostrar puntuacion
Dibujar pajaro
dibujar tuberias X
v
hay mas tuberias
f
hayQueAñadirTuberia
v
añadir nueva tuberia
f
actualizamos pajaro
actualizamos tuberias
calculamos colision
Eliminar tuberias fuera escena
pulsa tecla spacio
pajaro salta
pulsa tecla 'r'
reiniciar juego

Parte 4: Código fuente

Codifica el código fuente usando el lenguaje Java y la librería gráfica Processing/P5

```
class Pajaro{
  public float x=70;
  public float y=height/2;
  public float velocidad;
  public float aceleracion=0.6f;
  public PImage sprite;
  public boolean vivo = true;
  final static float CANTIDAD_SALTO = -10;

  public Pajaro(){
    sprite = loadImage("pajarito.png");
    sprite.resize(32,32);
  }

  void saltar(){
    velocidad = CANTIDAD_SALTO;
  }

  void dibujar(){

    final int ANCHO_PAJARO = 32;
    //fill(214,240,43);
    //ellipse(x,y, ANCHO_PAJARO, ANCHO_PAJARO); // CIRCULO

    push();
    // rotacion y translacion
    translate(x-ANCHO_PAJARO/2, y-ANCHO_PAJARO/2);
    rotate( radians(velocidad*2)   );
    image(sprite, 0, 0);
    pop();

  }

  void actualizarPosicion(){
    //velocidad es la derivada de la aceleración
    velocidad = velocidad + aceleracion;
    // posicion es la derivada de la velocidad
    y = y + velocidad;

    if(y > height) {
      velocidad = 0;
```

```
        y = height;
      }
    }
  }


  class Tuberia{
    public float x = width;
    public int espacioEntreTubos=150;
    public int posicionEspacio;
    final static float VELOCIDAD = 3;
    final static float ANCHO = 50;

    private int top, bottom;

    public Tuberia(){
          posicionEspacio = (int)random(espacioEntreTubos,
height-espacioEntreTubos);
          top = posicionEspacio - espacioEntreTubos/2;
          bottom = posicionEspacio + espacioEntreTubos/2;
    }
    void dibujar(){


      fill(0,255,0);
      //TODO 100 calcularlo
      rect(x,0,  ANCHO, top);

      //TODO 100 calcularlo
      rect(x, bottom, ANCHO, height-bottom);

      //Dibujamos las tapas
      rect(x-5, top-30, ANCHO+5+5, 30);
      rect(x-5, bottom, ANCHO+5+5, 30);

    }

    void actualizarPosicion(){
      x = x - VELOCIDAD;
    }

    boolean calcularColision(Pajaro pajaro){
      // calculo colision pajaro tuberia
      if(pajaro.y < top  ||  pajaro.y > bottom){
        if(pajaro.x > x && pajaro.x < x+ANCHO){
```

```
        return true;
      }
    }

   // calculo colision pajaro suelo
   if(pajaro.y >= height){
     return true;
   }

    return false;
  }
  boolean isFueraEscena(){
    if(x+ANCHO < 0){
      return true;
    }
    return false;
  }
}


class Fondo{
  PImage fondo;
  public Fondo(){
    fondo = loadImage("cielo.jpg");
  }
  void dibujar(){
    image(fondo, 0, 0);
    fondo.resize(width, height);
  }
}


class Suelo{
  PImage suelo;
  int posSuelo = 0;
  public Suelo(){
    suelo = loadImage("suelo.png");
  }
  void dibujar(){
    image(suelo, posSuelo, height - suelo.height);
    image(suelo, posSuelo + suelo.width, height - suelo.height);
    image(suelo, posSuelo + suelo.width * 2 , height - suelo.height);
  }
  void actualizarPosicion(){
    final int VELOCIDAD_SUELO = 3;
```

```
    // posicion es la derivada de la velocidad
    posSuelo = posSuelo - VELOCIDAD_SUELO;
    if(posSuelo < -suelo.width) {
      posSuelo = 0;
    }
  }
}

// Clase principal (ControlJuego)
Pajaro pajaro;
Fondo fondo;
Suelo suelo;
//TOFIX mejorar en el futuro cola, pila
ArrayList<Tuberia> tuberias;
int puntuacion;

void setup(){
  size(640,480);
  pajaro=new Pajaro();
  fondo = new Fondo();
  suelo = new Suelo();
  tuberias = new ArrayList<Tuberia>();
  puntuacion = 0;
}

void draw(){
  // borrar pantalla
  background(0);

  // dibujar fondo
  fondo.dibujar();


  // dibujar suelo
  suelo.dibujar();

  // mover suelo
  suelo.actualizarPosicion();

  // mostrar puntuacion
  fill(255,0,0);
  text(puntuacion, width/2, 35);

  // dibujar pájaro
  pajaro.dibujar();
```

```
  // dibujar tuberias
  for(int i=0;i<tuberias.size();i++){
    tuberias.get(i).dibujar();
  }

  // añadir tuberias nuevas
  final int FRECUENCIA_ANNADIDO_TUBERIAS = 100;
  if(frameCount % FRECUENCIA_ANNADIDO_TUBERIAS == 0){
    tuberias.add( new Tuberia() );
    puntuacion++;
  }

  // actualizamos posicion pajaro
  pajaro.actualizarPosicion();

  // actualizar posicion de tuberias
  for(int i=0;i<tuberias.size();i++){
    tuberias.get(i).actualizarPosicion();
  }

  // calcular colision
  for(int i=0;i<tuberias.size();i++){
    if(tuberias.get(i).calcularColision(pajaro) ){
      reiniciarJuego();
    }
  }

  // borrar tuberias fuera escena
  for(int i=0;i<tuberias.size();i++){
    if(tuberias.get(i).isFueraEscena()){
      tuberias.remove(i);
    }
  }

}

void reiniciarJuego(){
  puntuacion = 0;
  tuberias.clear();
  pajaro = new Pajaro();
}

void keyPressed(){
  if(key == ' '){
```

```
    pajaro.saltar();
  }
  if(key == 'r'){
    reiniciarJuego();
  }
}
```