

Proyecto: "Flappy Quetzal"

Versión 1.0

Descripción y Modelo de Análisis

Elaborado por:

- Cabello Díaz Sofía Elizabeth
- Lara Aguilar Christian Abraham
- Núñez Hernández Diego Ignacio
- Sánchez Hernández Carmen Alejandra

Fecha: 27/11/2020

1. Índice

1.	<i>Índice</i>	2
2.	<i>Introducción</i>	3
3.	<i>Casos de Uso-Análisis</i>	4
4.	<i>Diagrama de Clases</i>	5
5.	<i>Diagramas de Secuencia</i>	7
6.	<i>Diagramas de Estado</i>	8
7.	<i>Descripción de Arquitectura</i>	9

2. Introducción

El proyecto Flappy Quetzal es un juego de tipo "endless runner", que consiste en controlar un quetzal para hacerlo saltar y esquivar obstáculos hasta que choca con alguno.

Es un juego para un solo jugador incluye una interfaz gráfica donde el usuario puede mover al personaje principal (quetzal) hacia arriba y hacia abajo mientras el personaje vuela de manera constante.

La interfaz contiene obstáculos en forma de árbol. Cuando estos son tocados por el quetzal, muere. Cuando el jugador muere, el quetzal hace un sonido triste indicando el fin de la partida.

La principal funcionalidad que tiene el juego es un sistema de puntos que va aumentando conforme se van pasando los obstáculos y guarda los 5 mejores puntajes en un archivo de texto.

3. Casos de Uso-Análisis

Las funciones a las que puede acceder el usuario serán el acceso a los mejores puntajes y a crear una nueva partida cada vez que lo decida.



4. Diagrama de Clases

Desde el método Main (no mostrado en el diagrama) se llama a FlappyBird, esta clase será la encargada de instanciar todos los elementos necesarios para una partida.

Al instanciar pantallaDeJuego, crearPantalla pondrá todos los elementos en la pantalla de juego y se encargará de ir actualizándose, dependiendo del movimiento del jugador y de los obstáculos.

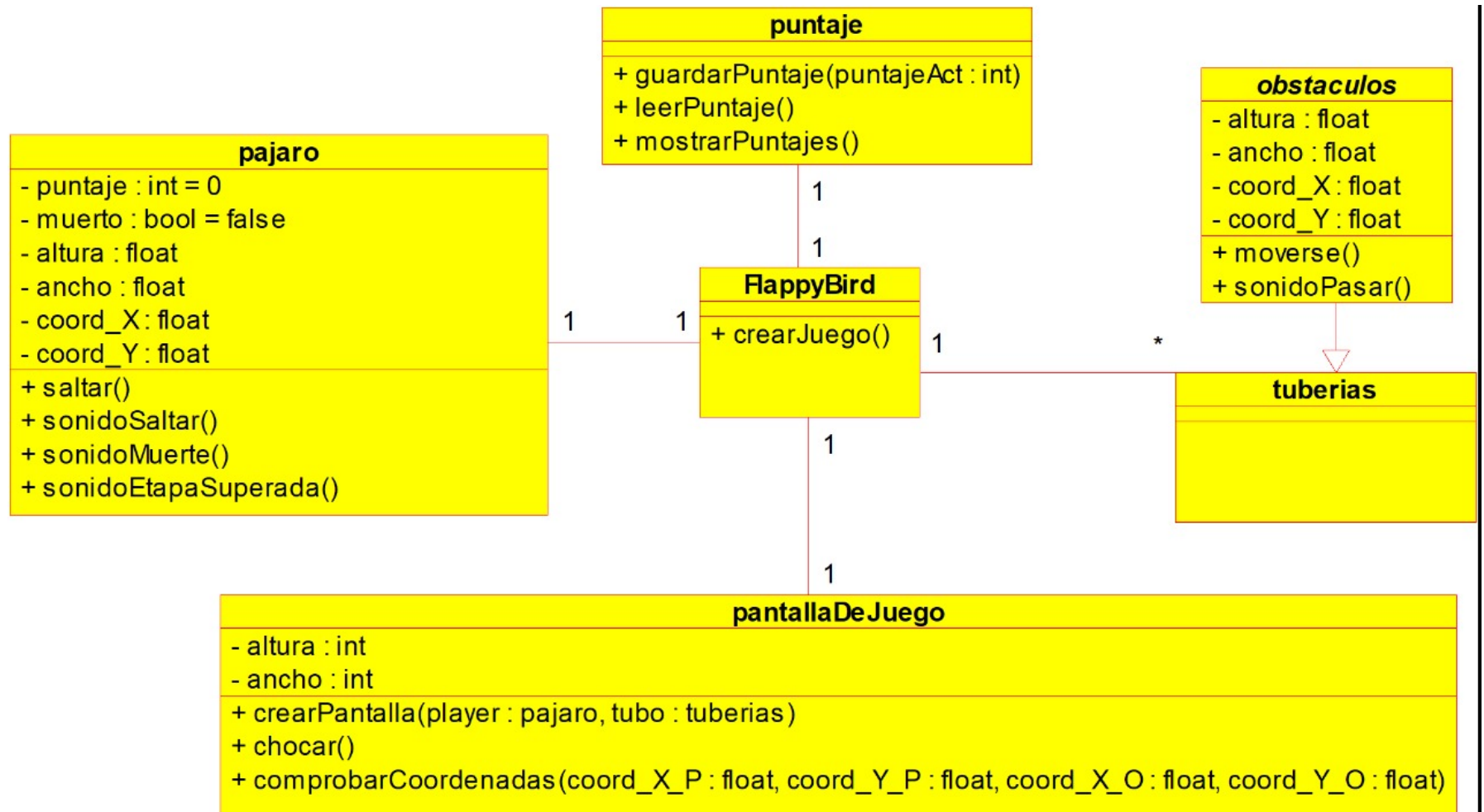
Cuando instanciamos pajaro, su puntaje iniciará en cero y muerto en false.

El método saltar modificará el atributo coord_Y conforme el jugador salte y se llamará al método sonidoSaltar.

La clase obstaculos será una clase abstracta de la cual las subclases (por ejemplo, tuberias) podrán heredar métodos y atributos. Esto se pensó con la idea de implementar más obstáculos en próximas etapas.

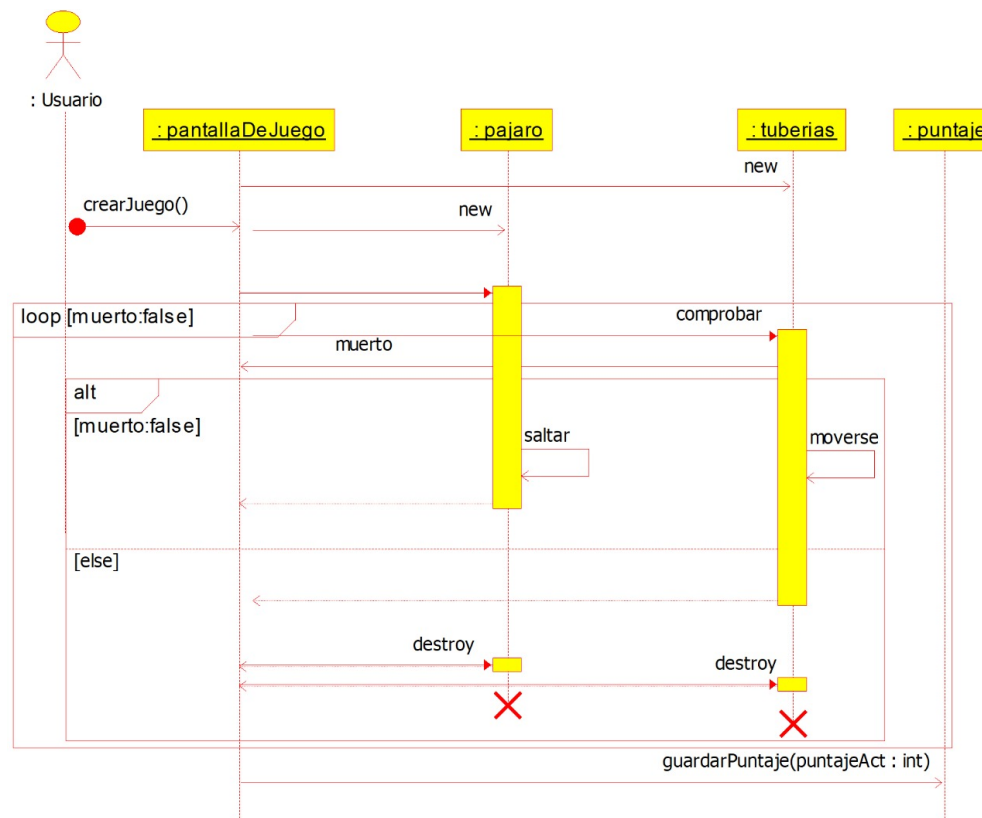
El método sonidoMuerte será llamado una vez que el método comprobarCoordenadas de la clase pantallaDeJuego verifique que las coordenadas del jugador (pajaro) y las coordenadas de algún obstáculo, sean las mismas. Una vez llamado al método sonidoMuerte, se cambiará el atributo muerto a true y se verificarán los puntajes previos mediante leerPuntaje de la clase puntaje, y dependiendo de si el puntaje nuevo obtenido en la partida es superior a los primeros tres puntajes almacenados en el archivo de donde se leerán los puntajes anteriores, se almacenará mediante guardarPuntaje.

Finalmente, se llamará a mostrarPuntajes y desde el método Main se preguntará si desea volver a iniciar una partida repitiendo todo el proceso descrito previamente.



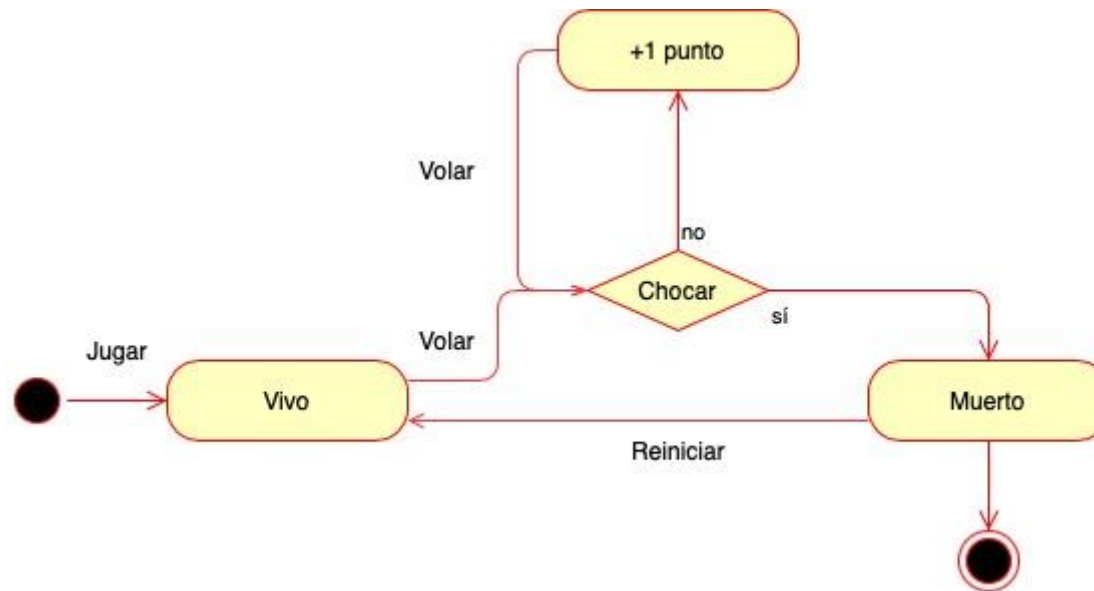
5. Diagramas de Secuencia

Este diagrama ilustra el momento en el que el usuario crea una partida. Primero se instancia un objeto de la clase pantallaDeJuego que a su vez instancia un objeto de tipo Pájaro y de tipo Tubería. Después se manda a llamar un método llamado “comprobar” que comprueba las coordenadas del Pájaro y de la Tubería para saber si el personaje ha chocado o no. En caso de no chocar, el pájaro sigue saltando y la tubería se sigue moviendo hasta que las coordenadas de los dos objetos coinciden y se acaba el juego guardando el puntaje.



6. Diagramas de Estado

En este diagrama se ilustran los estados del objeto Pájaro durante la ejecución del programa. Al inicio del juego, el pájaro está vivo por lo que podrá estar volando. Cuando el objeto toca un obstáculo pasa a un estado de muerto hasta que el usuario decida reiniciar el juego, mientras que no choque, su puntuación aumentará en un punto y seguirá volando.



7. Descripción de Arquitectura

El juego será ejecutado en una computadora que tenga instalada la máquina virtual de Java. Dentro de esta máquina virtual se ejecutará el archivo .jar del programa y a su vez hará uso de un archivo de texto para leer los puntajes del juego.

