

Laboratorio de programación

Trabajo Final Obligatorio 2019

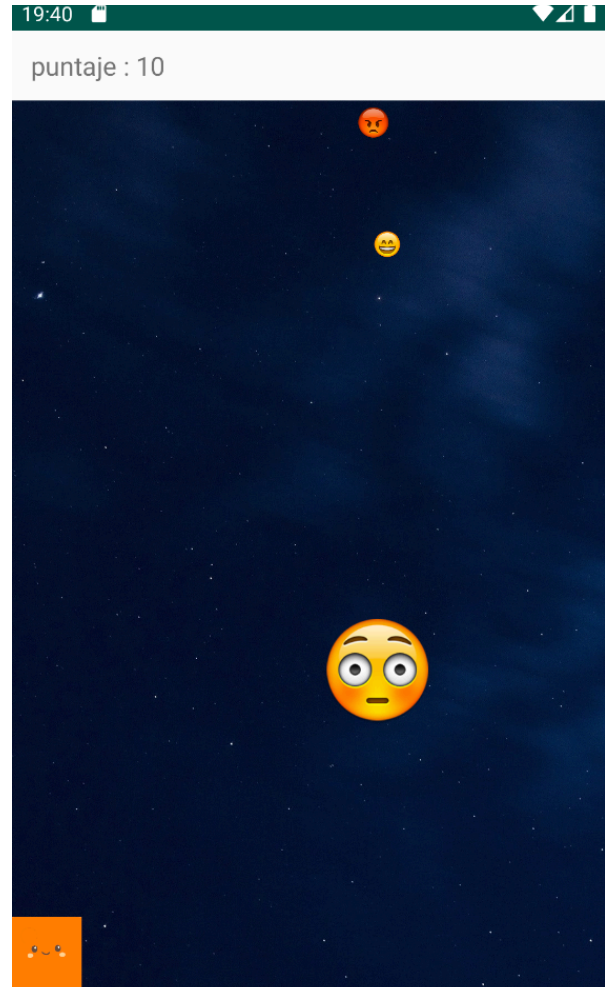
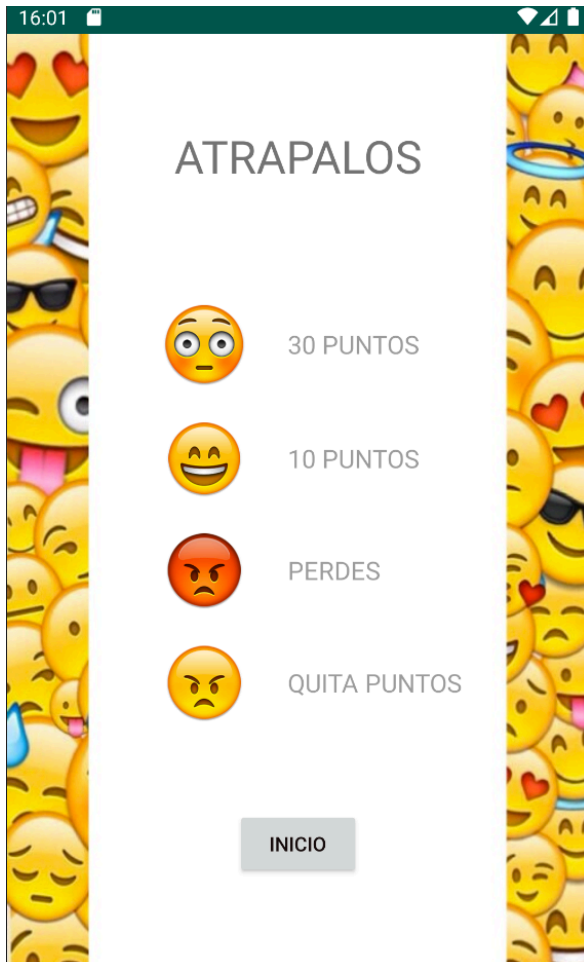
juego implementado en android studio : atrapalos

Ricardo Bauch FAI-1511



Introducción

El juego consiste en que el jugador gane la mayor cantidad de puntos atrapando emoticones que irán apareciendo a lo largo de la ejecución. Para ello el jugador debe desplazarse verticalmente y evitar ciertos emoticones “malignos “ que provocan que pierda el juego al primer impacto o pierda puntaje al atraparlos. A medida que se logre llegar a cierta cantidad de puntos la dificultad incrementa .

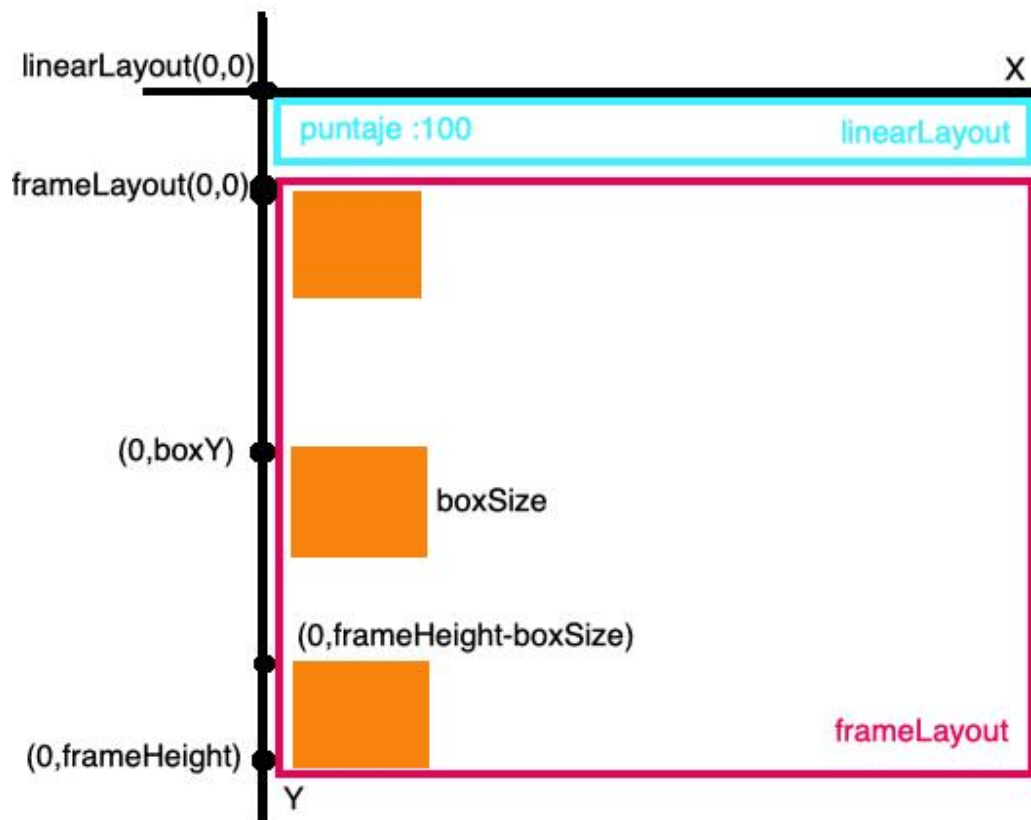


Implementación

- movimientos del box

Para el movimiento de la caja me guie del clásico juego “flappy bird” . Utilizó el método `onTouchEvent` para que cuando se presione la pantalla el box suba y al dejar de presionar baje , es por ello que hacemos el uso de `MotionEvent.ACTION_DOWN` y `MotionEvent.ACTION_UP`. Al realizar este metodo sucedió que al mantener presionado la pantalla , el box se iba fuera de rango al

igual cuando dejabamos de presionar y bajaba , es por ello que tuve que definir límites en la pantalla .



● Posicionando los emoticones

Para realizar los emoticones que irán desplazándose horizontalmente por la pantalla En el metodo `changePos` utilizó las coordenadas X e Y para posicionarlos fuera de la pantalla y que sus apariciones sean aleatorias . Los emoticones se van a desplazar decrementando el valor de X.

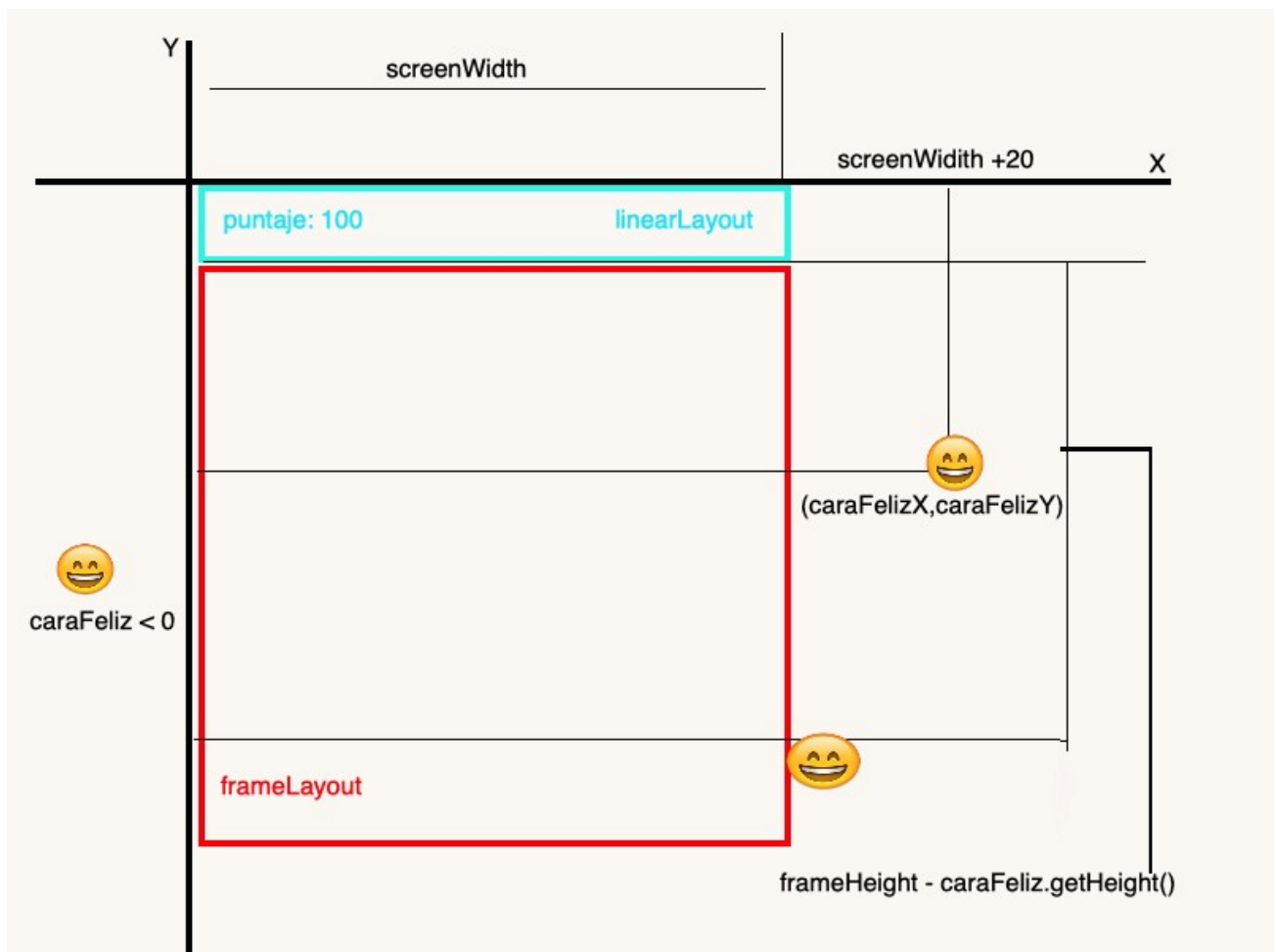
```
caraFelizX -= 12;
```

```
if(caraFelizX<0){
```

```
    caraFelizX = screenWidth +20;
```

```
    caraFelizY = (int)Math.floor(Math.random() * (frameHeight - caraFeliz.getHeight()));
```

```
}
```

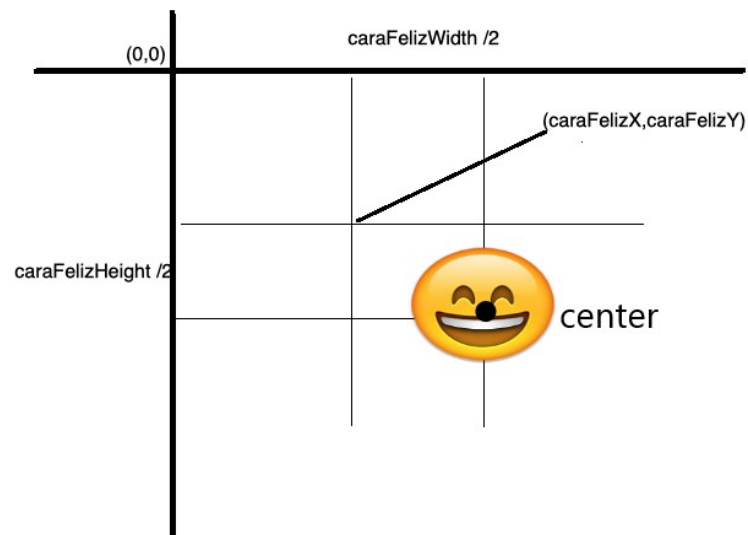


Verificando Impactos

Para el hit o impacto defini unas variables para que sirvan como centro del emoji así poder utilizarlas cuando toquen el box , todo esto lo codifique en el módulo ChekearImpacto.

```
int caraFelizCenterX = caraFelizX + caraFeliz.getWidth() / 2;
```

```
int caraFelizCenterY = caraFelizY + caraFeliz.getHeight() / 2;
```

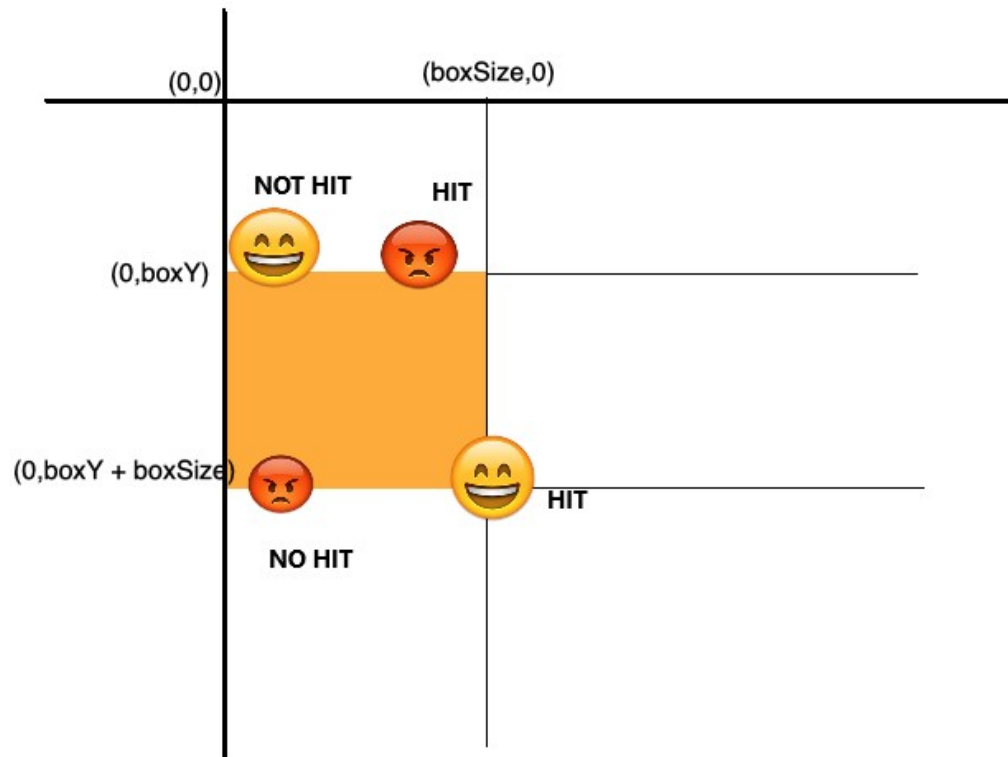


con las nuevas coordenadas del emoji y las coordenadas del box pude establecer el hit

```
0 <= caraFelizCenterX && caraFelizCenterX <= boxSize
```

```
boxY <= caraFelizCenterY
```

`caraFelizCenterY <= boxY + boxSize`



al chocar con el box se modifica el puntaje y ejecutó un sonido dependiendo del tipo de emoji que es.

`puntaje += 10;`

`caraFelizX = -10;`

`sonido.playHitSound();`

Si el emoji impactado fuera una “cara mala “ o roja como se ve en la imagen anterior el timer finaliza es decir estamos teniendo un evento de gameOver por lo tanto se inicia un activity de intent de la clase resultado

- Guardando el resultado

La idea del juego es ser competitivo con uno mismo por lo tanto decidí que cuando finalice el juego muestre el resultado de esa partida y el puntaje más alto obtenido en otras ejecuciones . Para implementar esto use una alternativa al SQLite propio de android, el “`SharedPreferences`” que se utiliza de manera muy sencilla.

Cada preferencia se almacenará en forma de clave-valor, y un valor asociado a dicho identificador .A diferencia de SQLite, los datos no se guardan en un fichero binario de base de datos, sino en ficheros XML.



● Sonido

Para los efectos de sonido utilizados en el impacto de los emoticones utilice la clase `soundPool`.

Mi metodo `playHitSound()` va a reproducir el sonido de impacto de un emoticon común en cambio `playOverSound()` reproduce el sonido provocado por un emoticon letal.

Al principio los efectos se escuchaban muy bajo en mi emulador por eso en el constructor de la clase agregue unas condiciones para tratarlo pero me di cuenta que era solo en el emulador ya que cuando lo instale en mi teléfono móvil todo funcionaba bien.

CONCLUSIÓN

Al realizar el trabajo ya estaba familiarizado con el entorno de trabajo de android studio gracias al otro TP obligatorio de la agenda que realizamos en el cursado por lo que dedique más tiempo en investigar los movimientos y las 2 clases que me fueron fundamentales para ponerlo a prueba, la clase handler y timer . Me gustaria que en el próximo cursado aparte del primer tp de android se realice otro tp similar en IONIC que permite realizar aplicaciones móviles de forma muy sencilla . Yo creo que lo importante de este trabajo fue ser capas de poner en practica la habilidad de investigar y aprender por nuestra cuenta ya que en el futuro la tecnologia va ir cambiando y con el tiempo nos vamos a tener que ir adaptando a ella.

MATERIAL DE INVESTIGACIÓN UTILIZADO

<https://sdesmedt.wordpress.com/2012/09/12/touch-handling-in-android/>

<https://developer.android.com/reference/android/os/Handler>

<https://developer.android.com/reference/java/util/Timer>

<http://www.sgoliver.net/blog/preferencias-en-android-i-shared-preferences/>

<http://www.androidcurso.com/index.php/recursos/37-unidad-6-multimedia-y-ciclo-de-vida/168-introduciendo-efectos-de-audio-con-soundpool>