Uso de Sprites y contadores

En este capítulo aprenderás a crear **DaleFuerte**, un juego inspirado en el clásico **Whac-A-Topo,** donde una serie de topos mecánicos aparecían por unos agujeros y los jugadores tenían que golpearlos con un mazo. **DaleFuerte** lo creó un miembro del equipo de **AppInventor** con la idea de probar el funcionamiento de un *sprite* que había implementado. Este término se usa en inglés para referirse a criaturas mitológicas, como hadas y duendes. Pero en la década de 1970 empezó a emplearse en la comunidad informática para hacer referencia a las imágenes capaces de moverse por la pantalla del ordenador (sobre todo, en los videojuegos). *Ellen Spertus* del equipo **AppInventor**, comenzó a trabajar con Sprites en la década de 1980.

Recuerda que iniciamos **AppInventor2** desde: http://ai2.appinventor.mit.edu/

¿ Qué vamos a construir ?

Para desarrollar la aplicación **DaleFuerte** necesitaremos:

- Un topo que se mueva **aleatoriamente** por la pantalla cambiando su posición cada segundo.
- Cuando el usuario lo toque, **el teléfono vibrará**, aumentarán los puntos y el topo saltará a una nueva posición.
- Si el usuario toca la pantalla pero no alcanza al topo, aumentará el **contador de fallos**.
- La aplicación tendrá un botón **Reset** (*Reiniciar*) que pondrá a cero los contadores de aciertos y fallos.

¿ Qué aprenderé ?

- El componente **ImageSprite**, que se utiliza con imágenes en movimiento sensibles al tacto en pantalla.
- El componente **Canvas**, que actúa como tablero de juego, donde colocaremos nuestro **ImageSprite**.
- El componente **Clock**, que determinará cuando se moverá el topo.
- El componente **Sound**, que producirá una vibración cuando el usuario golpee al topo.
- El componente **Button**, que iniciará un nuevo juego.
- **Procedimientos** para implementar comportamientos repetitivos, como el movimiento del topo.
- Generación de **números aleatorios**.

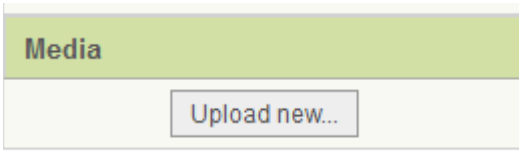
- Uso de bloques de **suma y resta.**

Empezamos

Conéctate al sitio web de **AppInventor** e inicia un nuevo proyecto. Denomínalo **DaleFuerte** (recuerda que los nombres de las aplicaciones en **AppInventor** no pueden tener espacios, aunque si los usas los sustituira por el carácter _). Asigna esta misma denominación al título de la pantalla.

Haz clic sobre el botón **Upload new...** que se encuentra en la sección **Media** (Multimedia) de **Components**.

Surgirá un cuadro de diálogo que te ayudará a seleccionar el archivo "*Mole.png"* en **AppInventor**.



Diseñar los componentes

Necesitaremos los siguientes componentes:

- **Canvas**: Hará las funciones de tablero de juego.
- **ImageSprite**: Muestra la imagen de un topo, la mueve por el lienzo y detecta cuándo se le toca.
- **Sound**: Hace vibrar el teléfono cuando el usuario toca al topo.
- **Labels**: Muestra los textos *Hits y Fallos.*
- **HorizontalArrangements**: se utiliza para definir la posición de *Labels*.
- **Button**: Se encargará de revisar los contadores de *aciertos y fallos* y dejarlos a 0.

- **Clock**: Será responsable de que el topo cambie su posición cada segundo.

La siguiente tabla muestra la lista de todos los componentes que intervienen en **DaleFuerte**:

Tipo de componente	**Grupo de Palette**	**Cómo lo llamaremos**	**Finalidad**
Canvas	**Drawing and Animation**	Canvas1	El contenedor de ImageSprite.
ImageSprite	**Drawing and Animation**	Topo	El usuario deberá intentar tocar el topo con el dedo.
Button	**User Interface**	**BotonReset**	El usuario tocará ese botón cuando quiera dejar a 0 los marcadores.
Clock	**Sensors**	Clock1	Controla el movimiento del topo.
Sound	**Media**	Sound1	Vibra cuando el usuario toca al topo.
Label	**User Interface**	Hits	Muestra Hits: en la pantalla.
Label	**User Interface**	ContadorHits	Muestra el número de aciertos.
Horizontal-Arrangement	**Layout**	Horizontal-Arrangement1	Coloca *HitsLabel* junto a *ContadorHits*.
Label	**User Interface**	Fallos	Muestra Fallos: en la pantalla.
Label	**User Interface**	ContadorFallos	Muestra el número de fallos.
Horizontal-Arrangement	**Layout**	Horizontal-Arrangement2	Coloca *la etiqueta Fallos* junto a ContadorFallos.

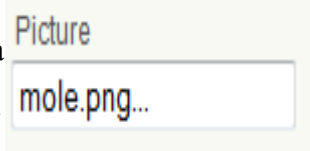
Colocar los componentes de acción

En esta sección vamos a colocar los componentes responsables de la acción del juego:

1. Arrastra un componente ***Canvas*** hasta **Viewer** y deja su nombre predeterminado ***Canvas1***. Ajusta el valor de la propiedad **Width** (ancho) a **Fill Parent** (Rellenar el elemento padre) para que tenga el mismo ancho que *Canvas*, es decir, el de la pantalla. Asigna a la propiedad

Height (alto) un valor de 300 píxels.

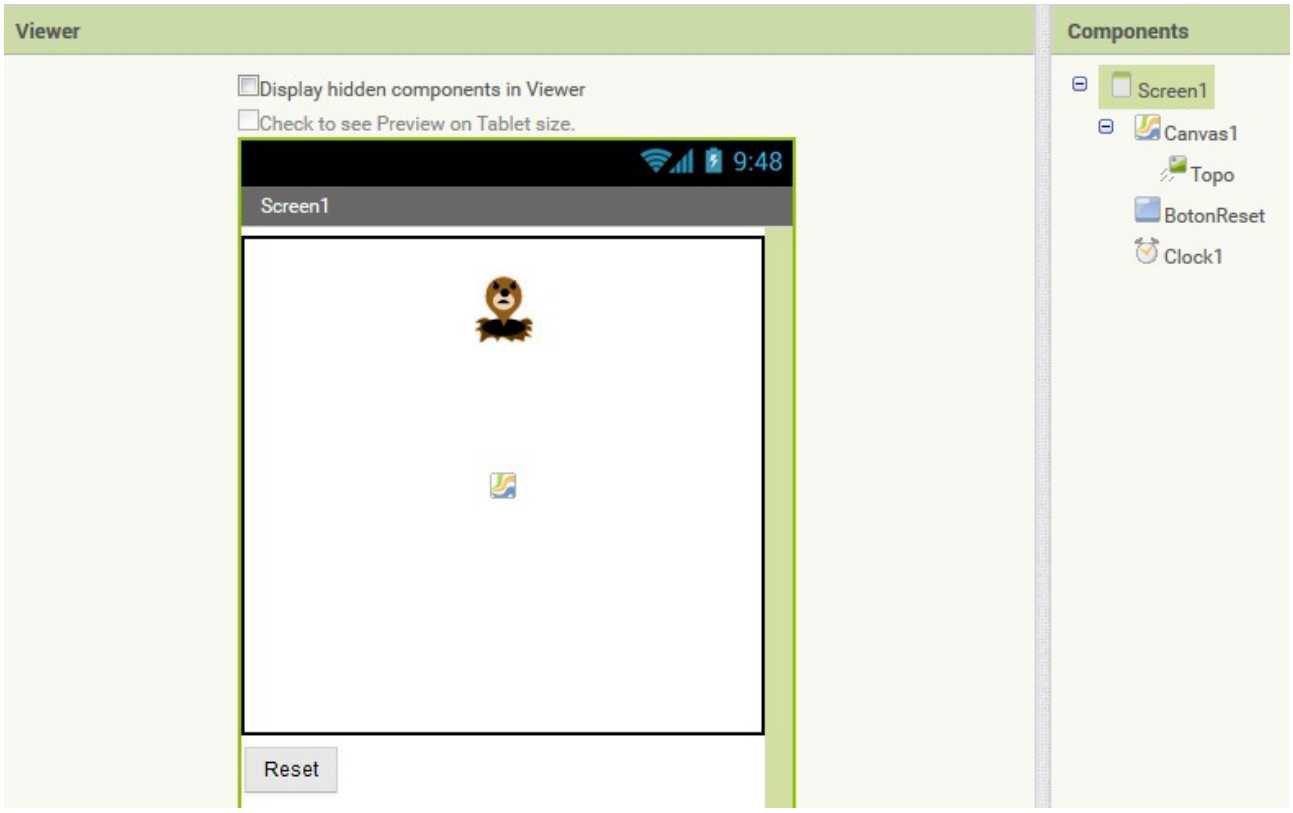
2. Mueve el componente ***ImageSprite*** y colócalo dentro de *Canvas1*. Haz clic sobre el botón **Rename** (Renombrar) que está debajo de la lista de **Components** y cambia su nombre por **Topo**. Pulsa sobre la propiedad **Picture** (imagen) para seleccionar el archivo "*Mole.png"*.



3. Desplaza el componente **Button** y colócalo debajo de *Canvas1*. Cambia su nombre a **BotonReset** y el contenido de la propiedad ***Text*** a **Reset**.

4. Arrastra el componente **Clock**. Aparecerá debajo de **Viewer**, en la sección reservada a los elementos invisibles: **non-visible components**.

La pantalla debería ser parecida a la de la figura, aunque es posible que el topo esté en otra posición.



Colocar los componentes Label

Vamos a poner los componentes encargados de mostrar la puntuación del usuario en la pantalla, en concreto, el número de *aciertos y fallos.*

1. Arrastra el elemento **HorizontalArrangement** del cajón **Layout** (Distribución de pantalla) hasta el área de trabajo y colócalo debajo del **BotonReset**. Deja su nombre predeterminado: **HorizontalArrangement1**.

2. Desplaza dos elementos **Label** y déjalos en **HorizontalArrangement1**.

    a) Cambia el nombre de la etiqueta (**Label**) de la izquierda por **Hits** y modifica su propiedad **Text** para que muestre el siguiente texto: **Hits:** ,dejando un espacio detrás de los dos puntos.

    b) Modifica el nombre de la etiqueta derecha por **ContadorHits** y escribe **0** en su propiedad **Text**.

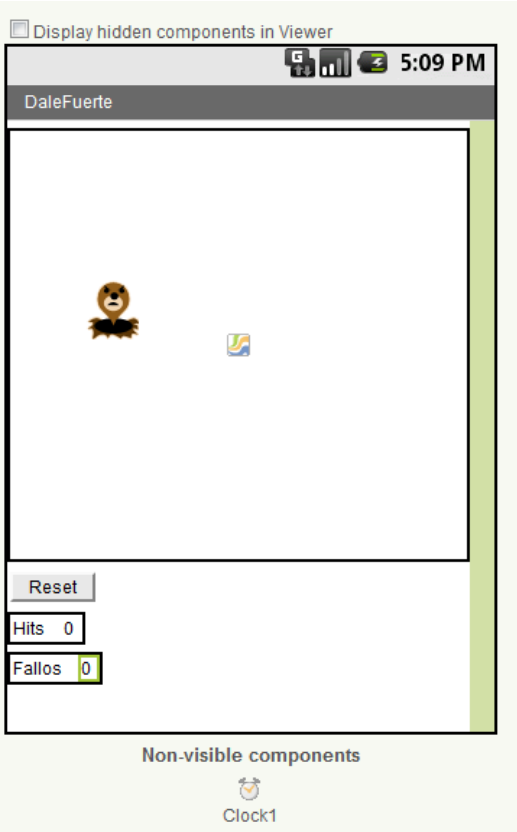
3. Arrastra un nuevo elemento **HorizontalArrangement** hasta el área de trabajo y colócalo debajo de **HorizontalArrangement1**. Deja su nombre predeterminado: **Horizontal Arrangement2**.

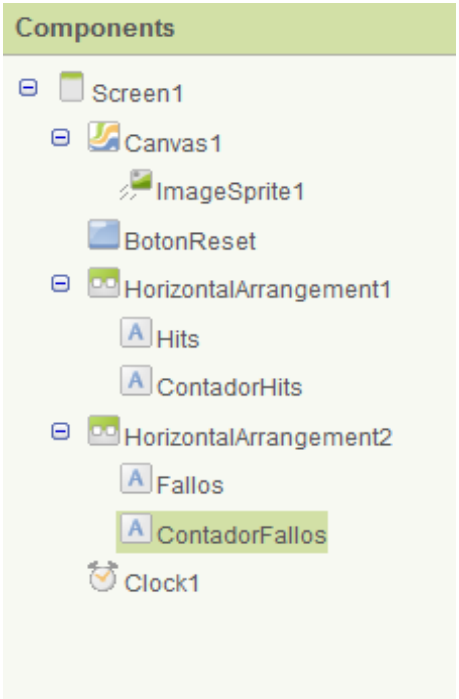
4. Mueve dos elementos **Label** hasta **HorizontalArrangement2**

    a) Cambia el nombre de la etiqueta (**Label**) de la izquierda por **Fallos** y modifica su propiedad **Text** para que muestre el siguiente texto: **Fallos:** , dejando un espacio detrás d ellos dos puntos.

    b) Modifica el nombre de la etiqueta derecha por **ContadorFallos** y escribe **0** en su propiedad **Text**.

La pantalla debería tener un aspecto similar al de la figura:





Asignar comportamientos a los componentes

Una vez creados los componentes, vamos a pasar a **Blocks** para definir sus comportamientos.

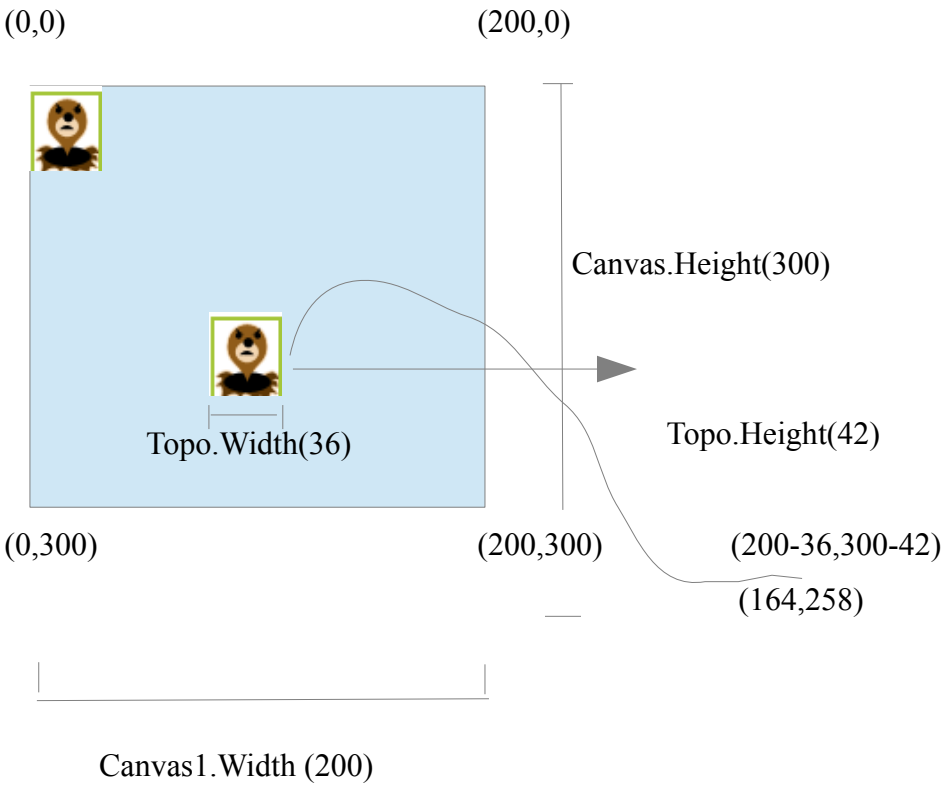
Mover el topo

En los programas vistos hasta la fecha hemos trabajado con procedimientos ya creados, como en el caso de **Vibrate** en *HolaGato*. ¿ No te gustaría que **AppInventor** tuviese uno que moviese el ***ImageSprite*** a una ubicación aleatoria ? Pues no lo tiene. La buena noticia es que podemos crearlo nosotros. Cuando lo hagamos, aparecerán dentro de un cajón y se podrán usar en cualquier aplicación. Le llamaremos **MoverTopo**. Lo invocaremos o llamaremos cuando se inicie el juego, si el usuario toca el topo y una vez cada segundo.

Crear MoverTopo

Para comprender cómo podemos mover el topo tenemos que saber algo sobre el funcionamiento de los gráficos en Android. Debemos ver el lienzo (y la pantalla) como una cuadrícula con coordenadas X (horizontales) e Y (verticales). Las de la esquina superior izquierda

serán (0,0). Cuando nos movamos hacia la derecha, iremos aumentando el valor de la coordenada X, del mismo modo que la Y crecerá según nos desplacemos hacia abajo. Las propiedades **X** e **Y** de **ImageSprite** indican dónde debería encontrarse la esquina superior izquierda. Así pues, los valores **X** e **Y** de un topo que se ubique en dicha esquina serán 0.



Para determinar el valor máximo que pueden tener las propiedades **X** e **Y** y que el topo no se salga de la pantalla tendremos que utilizar **Width** y **Height** de **Topo** y de **Canvas1**. Ambas deberán coincidir con la imagen del topo que vayamos a emplear.

Cuando creamos **Canvas1**, le asignamos una *altura* de **300 píxels** y al *ancho* de la pantalla el valor **Fill parent** que copia el ancho del elemento padre, en este caso, la pantalla (**Canvas**).Si el topo tiene *36 píxels de ancho* y el lienzo tiene *200*, la coordenada X del lado izquierdo del topo puede llegar a tener un valor mínimo de 0 (estará pegado al margen izquierdo) y máximo de 164 (*200-36* o, lo que es lo mismo, **Canvas1.Width - Topo.Width**), sin que el topo salga por el lado derecho de la pantalla. Del mismo modo, el valor mínimo de la coordenada Y será 0 y el máximo podrá ser **Canvas1.Height – Topo.Height**.

Para crear el procedimiento:

1. Desde el cajón **Built-In** (integrados) de **Blocks,** arrastra el bloque **to procedure arg do,** situado en **Procedures**, hasta el área de trabajo.

2. Haz clic sobre el texto **procedure** del bloque que acabas de colocar y sustitúyelo por **MoverTopo** como nuevo nombre del procedimiento.

3. Para mover el topo accede al contenido del cajón **Topo** y arrastra el elemento **Topo.MoveTo** al procedimiento. Suéltalo a la derecha de **do**. Fíjate que hay que asignar unas coordenadas **X e Y.**

4. Para especificar que el valor de la coordenada X debe estar comprendido entre **0** y **Canvas1.Width – Topo.Width**, como especificamos antes:
    a) Haz clic en la pestaña **Built-In** para acceder a los procedimientos. Pulsa en el cajón **Math**.
    b) Arrastra el bloque **random integer** y suéltalo en el área de trabajo, de tal forma que el saliente que se encuentra a la izquierda encaje en la ranura **X** de **Topo.MoveTo**.

    c) Cambia el número **1** de la ranura **from** por **0**. Para ello, deberá hacer clic sobre él.

    d) Elimina el valor **100**. Pulsa sobre él y presiona **Supr** del teclado.También puedes arrastrar el valor hasta la papelera del área de trabajo.

    e) Haz clic sobre el cajón **Math** y desplaza el bloque de la **resta (-)** hasta la ranura **to**.

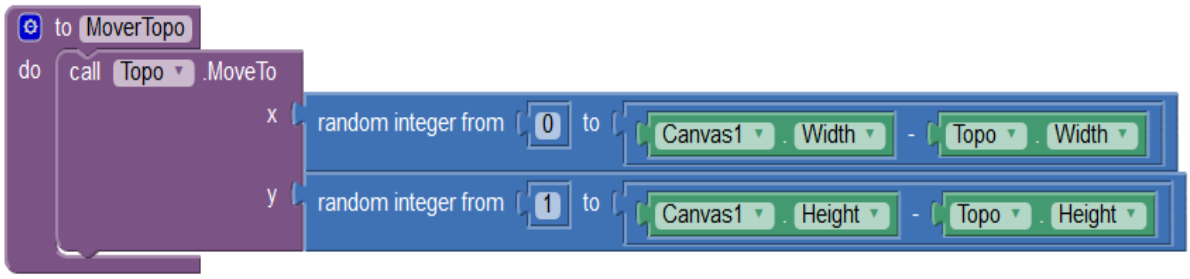
    f) Haz clic en el cajón **Canvas1** y desplázate por la lista hasta el elemento **Canvas1.Width**. Arrástralo hasta el lado izquierdo del bloque de la resta que puso en el área de trabajo en el paso anterior.

    g) Abre el cajón **Topo** y mueve el elemento **Topo.Width** hasta el lado derecho del

bloque de la resta.

5. Sigue el mismo procedimiento para especificar que el valor de la coordenada **Y** debe ser uno aleatorio entre **0** y **Canvas1.Height – Topo.Height**.

Deberás haber hecho los siguientes bloques:



Guarda primero el proyecto con el nombre **DaleFuerte**, se generará el fichero **"DaleFuerte.apk"** que es el que debes copiar a tu teléfono y probarlo ahora.

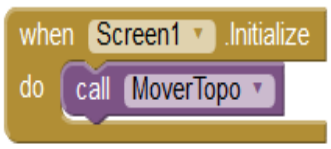
PRUEBA!!!!

Llamar a MoverTopo cuando se inicia la aplicación

Ahora vamos a utilizar el procedimiento **MoverTopo**. Queremos que la aplicación haga algo cuando se abra, para ello recurriremos al bloque **Screen1.Initialize**

1. Haz clic sobre el cajón **Screen1** y arrastra **Screen1.Initialize** hasta el área de trabajo.

2. Abre el cajón **Procedures** y localiza el bloque llamado **call MoverTopo**. ¿ Te has fijado ? Es el bloque que acabas de crear. Arrástralo hasta **Screen1.Initialize**.

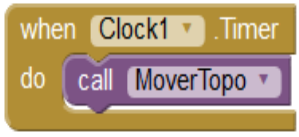


Llamar a MoverTopo cada segundo

Para hacer que el topo se mueva cada segundo necesitaremos el componente **Clock** (Reloj). Dejaremos el valor predeterminado (1000) de la propiedad **TimerInterval** o, lo que es lo mismo, un segundo. Es decir, cada segundo tendrá lugar aquella acción que hayamos especificado en **Clock1.Timer**. Vamos a configurarlo:

1. Abre el cajón **Clock1** y arrastra **Clock1.Timer** hasta la zona de trabajo.

2. Pulsa en el cajón **Procedures** y arrastra el bloque **call MoverTopo** hasta **Clock1.Timer**, como en la figura:



Si el juego le parece demasiado fácil, prueba a cambiar el valor de la propiedad **TimerInterval** de **Clock1** para hacer que el topo se desplace más a menudo. Acuérdate que debes modificar esto desde **Components Designer**.

La puntuación

Hemos creado dos etiquetas con valores iniciales 0 para los aciertos y los fallos: **ContadorHits** y **ContadorFallos.** Ahora vamos a incrementarlos cuando el usuario consiga alcanzar al topo (habrá tenido éxito) o cuando falle (contará como un error). Para ello, emplearemos el bloque **Canvas1.Touched** que indica que el usuario ha tocado la pantalla, las coordenadas X e Y del punto donde lo ha hecho (que no nos hará falta conocerlo) y si alcanzó al topo (que tampoco necesitamos saberlo).

Cada vez que se toque la pantalla, la aplicación comprobará si se ha alcanzado al topo. Cómo sólo habrá un topo (**sprite**) en el programa, tiene que ser **Topo1**. Si el usuario ha tocado a *Topo1*, la aplicación incrementará en un número el contador de aciertos (**ContadorHits.Text**). Pero si no se ha conseguido cazarlo, aumentará en una unidad el contador de fallos (**ContadorFallos.Text**). Si el usuario no logra tocarlo, el valor de **touchedSprite** será false.

Vamos a crear estos bloques:

1. Abre el cajón **Canvas1** y arrastra el elemento **Canvas1.Touched** hasta el área de trabajo.

2. En **Built-In**, abre el cajón **Control** y arrastra el elemento **ifelse** hasta **Canvas1.Touched.**

3. Pulsa sobre **Variables** y desplaza el elemento **get** hasta la ranura **test** de **ifelse**. Despliga el control get y elige **touchedAnySprite.**

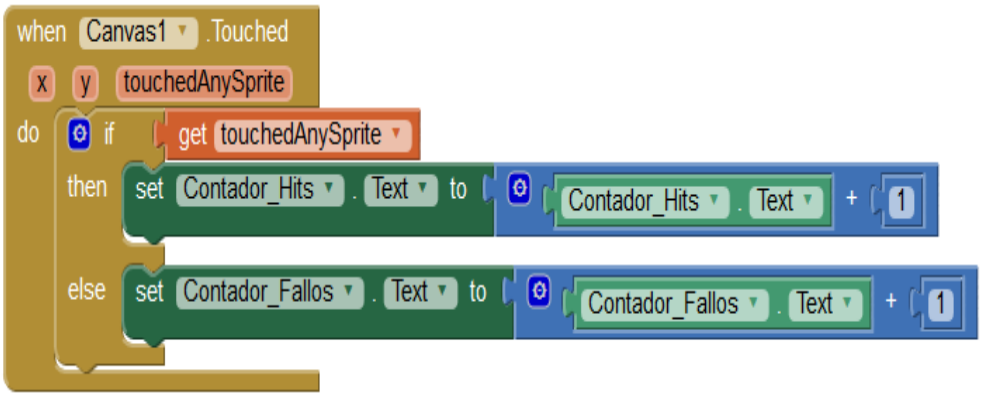
4. Como queremos incrementar el valor de **ContadorHits.Text** cada vez que el usuario tenga éxito golpeando al topo:
   a) Abre el cajón **ContadorHits** y arrastra el bloque **setContadorHits.Text to** hasta colocarlo a la derecha de **then-do**.

   b) Accede ahora al contenido de **Built-In**, abre el cajón **Math** y mueve el sigo de la **suma (+)** hasta la ranura **to**.

   c) Accede al cajón **ContadorHits** y desplaza el bloque **ContadorHits.Text** hasta el lado izquierdo de la suma.

   d) Vuelve a **Built-In**, abre el cajón **Math** y arrastra el bloque **numerico** hasta la posición libre que se encuentra a la derecha del signo + . Haz clic sobre el número y cámbialo por 1.

5. Repite el paso 4 para completarlas operaciones de **ContadorFallos**. Deberás colocarlas en la ranura **else-do**. Obtendrás:



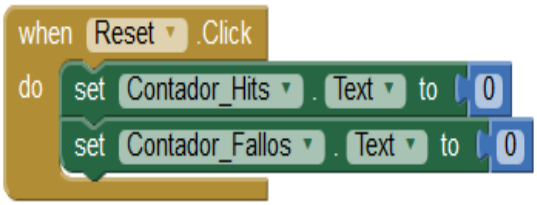
Guarda primero el proyecto con el nombre **DaleFuerte2**, se generará el fichero **"DaleFuerte2.apk"** que es el que debes copiar a tu teléfono y probarlo si funciona correctamente ahora.

PRUEBA!!!!

Ya puedes probar el funcionamiento del nuevo código en tu teléfono. Toca la pantalla e intenta golpear al topo. Cuando aciertes, se incrementará el valor del contador de aciertos. Si no lo haces, crecerá el contador de fallos. En capítulos posteriores aprenderemos a crear procedimientos más potentes: sumar argumentos, proporcionar resultados y establecer llamadas por sí mismos.

Reiniciar los marcadores de puntuación

Para empezar una nueva partida necesitaremos borrar las puntuaciones y dejar los marcadores a 0. Para ello, necesitaremos un bloque **BotonReset.Click** que asigne el valor 0 a **ContadorHits.Text** y **ContadorFallos.Text**.



Guarda primero el proyecto con el nombre **DaleFuerte3**, se generará el fichero **"DaleFuerte3.apk "**que es el que debes copiar a tu teléfono y probarlo ahora.

PRUEBA!!!!

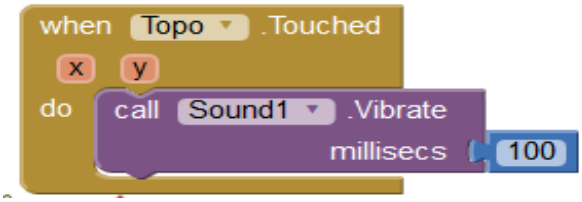
Trata de cazar al topo y observa como aumentan los contadores de aciertos y fallos. Luego, presiona el botón **Reset** para dejarlos a 0.

Qué debe hacer la aplicación cuando cazamos al topo

Antes hemos comentado que queríamos que el teléfono vibrase cuando cazamos al topo. Para ello, debemos utilizar el bloque **Sound1.Vibrate.**

- Ves al **Topo** y desplaza **Topo.Touched** al área de trabajo.

- Falta agregar el sonido. Ves a **Component Designer**, sección **Media** y arrastra el elemento **Sound** al área de trabajo. Ahora, desde **Blocks** vemos que ya aparece el elemento **Sound1**, lo pinchamos y arrastramos su función **Vibrate** dentro del bloque **Topo.Touched**. Completa el número de **milisegundos** para que el teléfono vibre durante **100 ms** cada vez que cazamos al topo.



Observa cómo vibra el teléfono cada vez que atrapas al topo. Si te parece que lo hace demasiado tiempo, modifica el número de **milisegundos** de la propiedad **Sound1.Vibrate**.

Guarda primero el proyecto con el nombre **DaleFuerte4**, se generará el fichero **"DaleFuerte4.apk "**que es el que debes copiar a tu teléfono y probarlo ahora.

PRUEBA!!!!

Variaciones

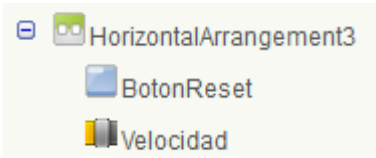
Aquí te dejo algunas propuestas para mejorar **DaleFuerte**:

1. Añade botones que permitan al usuario **aumentar la velocidad de movimiento del topo**.

2. Incorpora una etiqueta que informe del número de veces que se ha movido el topo, es decir, que muestre **cuántas ocasiones ha aparecido** en la pantalla.

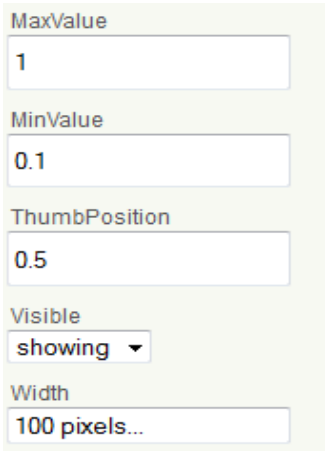
3. Introduce **un segundo ImageSprite** con la imagen de algo que el usuario no deba tocar, como por ejemplo, una **TopoMalo("Mole2.png")**. Si la toca, la aplicación la penalizará quitándole un punto de su contador de aciertos o incluso finalizando el juego.

La solución es:

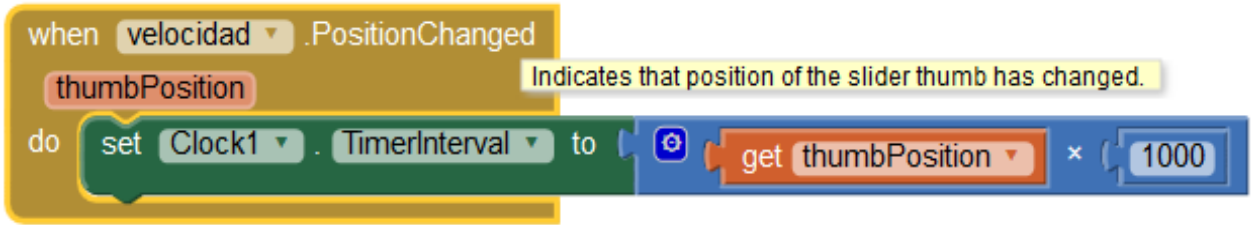


1. Añadiremos un deslizador (**Slider**)que colocaremos dentro de un elemento de disposición horizontal(**Horizontal Arrangement)** para controlar la velocidad.

Dentro del Slider establece estos valores: *Maximo, minimo y medio*. Establece como ancho **(Width)** *100 pixels*

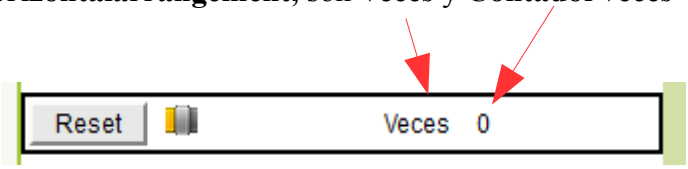


En el **Blocks** establece el siguiente código:

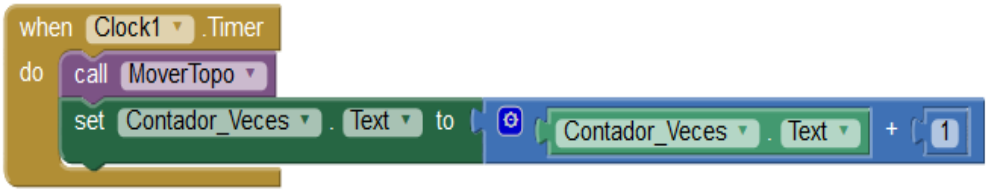


La velocidad variará entre 0,1*1000 ms (100ms) y 1*1000 ms.

2. Arrastra dos etiquetas (**Labels**) dentro del **Horizontalarrangement,** son **Veces** y **ContadorVeces**



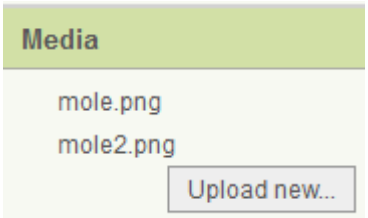
Establece como campo **Text** de **ContadorVeces** 0 y dentro del **Blocks Editor** establece este código.



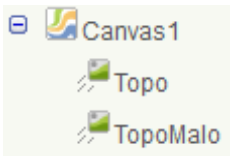
3. Para esta modificación hay que cambiar parte del código asociado a **canvas1.touched**, ya que ahora tenemos **dos sprites** y añadir código para cuando esos sprites sean tocados.

Así pues:

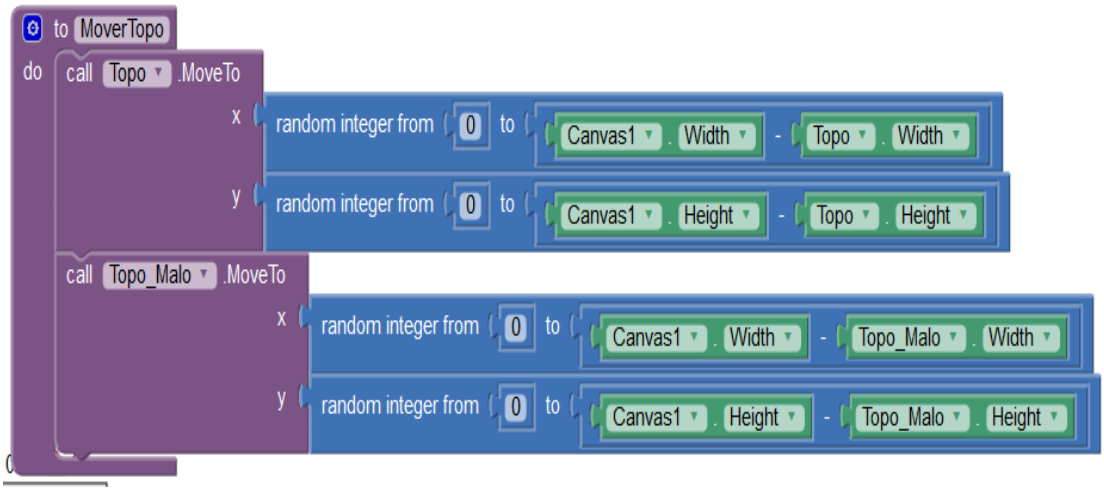
- Carga dentro de **Media** el nuevo objeto, *"Mole2.png"*:



- Arrastra desde **Drawing and Animation** otro **imageSprite** a Canvas y renómbralo como **TopoMalo**



- Modifica **MoverTopo** para que se encargue del movimiento de los dos topos.



- Los códigos asociados a **Tocar Topo**, **Tocar TopoMalo** y **Tocar Canvas** son: