



App Inventor

PRÁCTICA 5. MoleMash



En esta práctica se muestra cómo crear **MoleMash**, un juego inspirado en el clásico arcade **Whac-A-Mole**, en el que criaturas mecánicas salían brevemente fuera de unos agujeros, y los jugadores conseguían puntos al golpearlos con un mazo. MoleMash fue creado para probar la funcionalidad de los **Sprite**.

1.- Lo que vas a construir

En la aplicación MoleMash que se muestra en la Figura 1, podrás ver las funciones siguientes:

- Un topo aparece en lugares al azar en la pantalla, moviéndose una vez por segundo.
- Tocar un topo hace que el dispositivo vibre, la puntuación se incrementará (aumenta en uno), y el topo se moverá inmediatamente a una nueva ubicación.
- Al tocar la pantalla, pero en un espacio vacío, hará que la puntuación de los fallos se incremente.
- Al pulsar el botón Restablecer, restablecerá los puntos de aciertos y errores.



2.- Lo que aprenderás

El tutorial incluye los siguientes componentes y conceptos:

- El componente **ImageSprite** para imágenes móviles táctiles.
- El componente **Canvas**, que actúa como una superficie sobre la que colocar el **ImageSprite**.
- El componente **Clock (Reloj)** para mover el sprite en un tiempo determinado.
- El componente **Sound** para producir una vibración cuando se toca el topo.
- El componente **Button** para iniciar un nuevo juego.
- **Procedimientos** para implementar un **comportamiento repetido**, como mover el topo.
- La generación de números aleatorios.
- Uso de la adición (+) y de resta (-) bloques.

3.- Primeros pasos

Conéctate al sitio **Web de App Inventor** y empieza un **nuevo proyecto**. El nombre será "**MoleMash**", y también establece el título de la pantalla para "**MoleMash**". Abre el **Editor de bloques** y conecta con el **emulador** o el **móvil**.

Descarga la imagen del **topo en el agujero (mole.png)** y súbela a **AppInventor**.





4.- Diseño de los componentes

Vamos a usar estos componentes para hacer **MoleMash**:

- Un **Canvas** que sirve como un campo de juego.
- Un **ImageSprite** que muestra una imagen de un topo y puede moverse y sentir cuando se toca.
- Un **Sound** que vibra cuando se toca el topo.
- **Labels** que muestran "**Impactos**" i "**Errores**", y los números reales de aciertos y errores.
- **HorizontalArrangements** para posicionar correctamente las **Labels** .
- Un **Button** para restablecer el número de aciertos y errores a "0".
- Un **Clock (reloj)** para que el topo se mueva una vez por segundo.

La tabla 2 muestra la lista completa de los componentes.

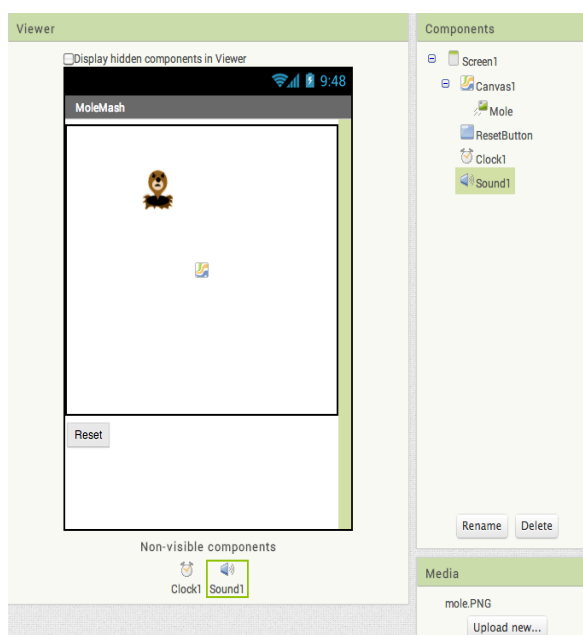
Tabla 2. La lista completa de componentes para MoleMash

Component type	Palette group	What you'll name it	Purpose
Canvas	Basic	Canvas1	The container for ImageSprite.
ImageSprite	Animation	Mole	The user will try to touch this.
Button	Basic	ResetButton	The user will press this to reset the score.
Clock	Basic	Clock1	Control the mole's movement.
Sound	Media	Sound1	Vibrate when the mole is touched.
Label	Basic	HitsLabel	Display "Hits: ".
Label	Basic	HitsCountLabel	Display the number of hits.
Horizontal-Arrangement	Screen Arrangement	HorizontalArrangement1	Position HitsLabel next to HitsCountLabel.
Label	Basic	MissesLabel	Display "Misses: ".
Label	Basic	MissesCountLabel	Display the number of misses.
Horizontal-Arrangement	Screen Arrangement	HorizontalArrangement2	Position MissesLabel next to MissesCountLabel.

5.- La colocación de los componentes de la acción

En esta sección, vamos a colocar los componentes necesarios para la acción del juego.

1. Arrastra un **lienzo** , dejándolo con el nombre predeterminado **Canvas1** . Establece su **Ancho a "Fill Parent..."** por lo que será tan ancho como la pantalla, y ajusta su **altura** a 300 píxeles.
2. Arrastra un componente **ImageSprite** del grupo de la paleta de **Drawing and Animation**. Colócalo en cualquier parte de **Canvas1** . Haz clic en cambiar el nombre en la parte inferior de la lista de **Componentes** y **cambia su nombre a "Mole"**. Establece su propiedad **Image** a **mole.png** , que habrás subido antes.
3. Arrastra un **botón** , colocándolo debajo de **Canvas1** . Cambia el nombre a **"ResetButton"** y establece su propiedad de **texto** en **"Reset"**.
4. Arrastra en un componente **Clock (reloj)**. Aparecerá en la parte inferior del visor en la sección **"Los componentes no visibles"**.
5. Arrastra una **Sound** . También aparecerá en la sección **"Los componentes no visibles"**. Tu pantalla debe parecerse a la Figura 3 (a pesar de que tu topo puede estar en una posición diferente).

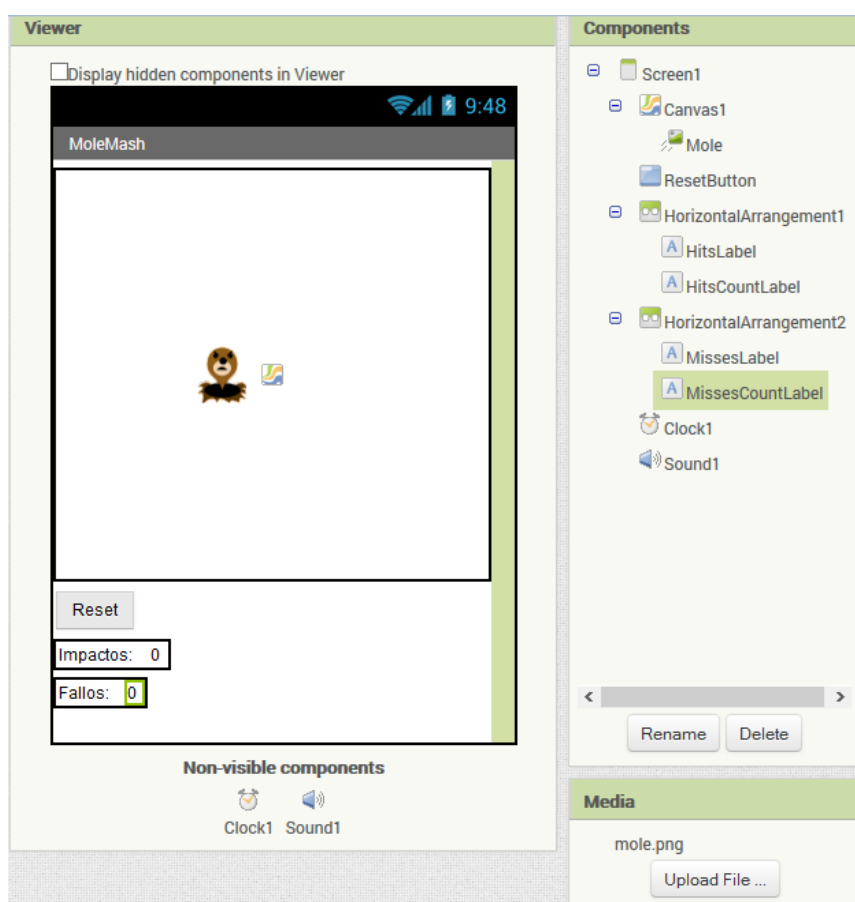




6.- Colocación de los componentes Label

Ahora vamos a colocar los componentes para la visualización del usuario del número de aciertos y errores.

1. Arrastra un **HorizontalArrangement** , colocándolo debajo de **Button** y manteniendo el nombre predeterminado de **HorizontalArrangement1** .
 2. Arrastra dos **etiquetas** a **HorizontalArrangement1** .
 - Cambiar el nombre de la **etiqueta izquierda** a "**HitsLabel**" y establece su **texto** a "**Impactos:** " (asegurándose de incluir un espacio después de los dos puntos).
 - Cambiar el nombre de la **etiqueta derecha** a "**HitsCountLabel**" y establece su **texto** a "**0**".
 3. Arrastra un segundo **HorizontalArrangement** , colocándolo debajo **HorizontalArrangement1** .
 4. Arrastra dos etiquetas en **HorizontalArrangement2** .
 - Cambiar el nombre de la **etiqueta izquierda** a "**MissesLabel**" y establece su **texto** a "**Fallos:** " (asegurándose de incluir un espacio después de los dos puntos).
 - Cambiar el nombre de la **etiqueta derecha** de "**MissesCountLabel**" y establece su **texto** a "**0**".
- Tu pantalla debería parecerse a:.



La vista Designer de todos los componentes MoleMash

7.- Programar los comportamientos a los Componentes

Después de crear los componentes anteriores, pasamos al Editor de bloques para programar el comportamiento del programa. En concreto, queremos que el topo se mueva a un lugar al azar en el lienzo a cada segundo.

El objetivo del usuario es pulsar en el topo siempre que aparezca, y la aplicación mostrará el número de veces que tenemos éxito en los impactos y las veces que no lo alcanzamos. (Nota: "¡Se recomienda el uso de tu dedo, no un mazo!" ;-)) Al presionar el botón "Reset", restablece el número de aciertos y errores a 0.

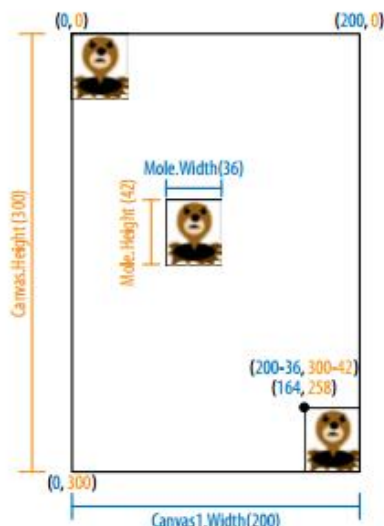
8.- Movimiento del topo

En los programas que has realizado hasta ahora los procedimientos, como la **vibración** en HelloPurr estaban preestablecidos. No existen procedimientos establecidos para todos los eventos que podamos realizar, como hacer que se mueva un a imagen por la pantalla de forma aleatoria, pero sí podemos crear nuestros propios procedimientos. Al igual que los procedimientos incorporados, tus procedimiento se mostrará en una paleta y se puede utilizar en cualquier lugar de la aplicación.

En concreto, vamos a crear un procedimiento para mover el topo a un lugar al azar en la pantalla, que denominaremos **MoveMole**. Queremos llamar a **MoveMole** en el **inicio del juego**, cuando el usuario **toca con éxito el topo**, y **una vez por segundo**.

9.- Creación del procedimiento "MoveMole"

Para entender cómo **mover el topo**, tenemos que ver cómo funcionan los **gráficos Android**. El **lienzo** (y la pantalla) se pueden considerar como una cuadrícula con **coordenadas X (horizontal) e Y (vertical)**, en el que el las coordenadas **(x, y)** de la esquina superior izquierda son **(0, 0)**. La **coordenada X** aumenta a medida que se mueve hacia la derecha, y la **coordenada Y** aumenta a medida que se mueve hacia abajo, como se muestra en la **Figura 5**. Las propiedades **X** e **Y** de un **ImageSprite** indican donde estará su esquina superior izquierda, por lo que el topo, cuando está en la esquina superior izquierda tiene valores **X** e **Y** de **0**.



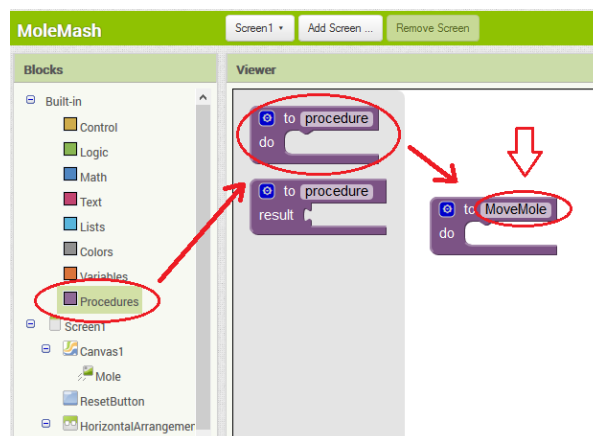
Para determinar los **valores máximos disponibles de X e Y** para que "**Mole**" (el topo) se vea en la pantalla, tenemos que hacer uso de la propiedades **Anchura** y **Altura** de **Mole** y **Canvas1**.

(Las **propiedades Anchura** y **Altura** del topo son las mismas que el tamaño de la imagen que has subido. Cuando creaste **Canvas1**, ajustaste su **altura** a 300 píxeles y su **anchura** a "**Fill parent...**", que copia la anchura de la pantalla.).

Si el topo es de **36 píxeles de ancho** y el **lienzo es de 200 píxeles de ancho**, la **coordenada X** de la parte izquierda del topo puede ser tan baja como **0** (hasta el final a la izquierda) o tan alta como **164** ($200-36$, o **Canvas1.Width - Mole.Width**) si el topo que se mueve al borde derecho de la pantalla. Del mismo modo, la **coordenada Y** de la parte superior del topo puede variar de **0** a **Canvas1.Height - Mole.Height**.

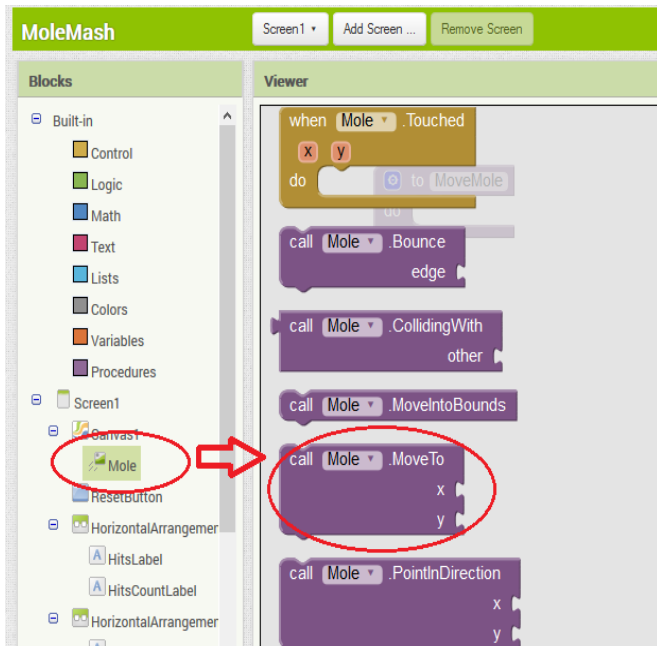
Posiciones del topo en la pantalla.

Crearemos el procedimiento arrastrando un bloque de **procedimiento** desde la paleta **Build-it** y asignando el nombre de "**MoveMole**" :





Ahora indicaremos **qué hace el procedimiento, mover el Sprite del topo**. Arrastraremos un bloque desde la paleta de la imagen del topo:



Para colocar **al azar el topo**, tendremos que seleccionar una **coordenada X** en el **rango de 0 a Canvas1.Width - Mole.Width**. Del mismo modo, vamos a establecer la **coordenada Y** para estar en el **rango de 0 a Canvas1.Height - Mole.Height**. Podemos generar un **número aleatorio** a través del **procedimiento integrado "random integer"** (entero aleatorio), que se encuentra en la paleta de **Matemáticas**.

Tendrás que cambiar la configuración del parámetro predeterminado **"from"** de **1 a 0** y definir el parámetro **"to"**, como se muestra en la **Figura 6**.

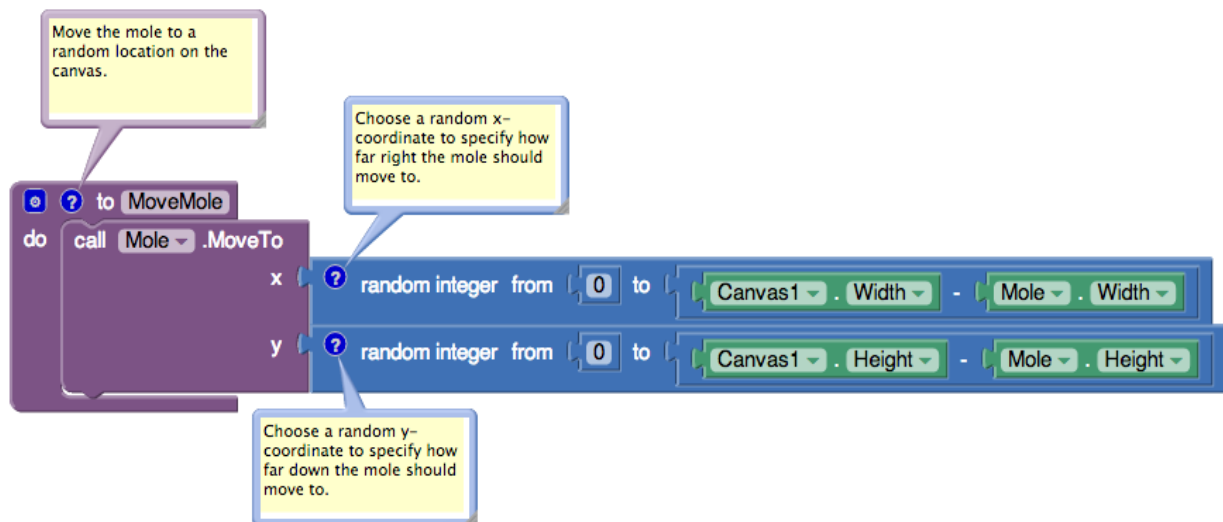
Desecha el **número 100** haciendo clic en él y pulsando **Del** en el teclado o el botón Eliminar o arrastrándolo a la papelera.

Haz clic en la paleta de **Matemáticas** y arrastra un **bloque de resta (-)** al conector **"to"**.

Haz clic en paleta de **Canvas1** y arrastra un bloque **Canvas1.Width** hacia el lado izquierdo de la operación de resta.

Del mismo modo, haz clic en la paleta **Mole** y arrastra **Mole.Width** al espacio de trabajo colócalo en el lado derecho del bloque de la resta.

Repite las operaciones para especificar que la **coordenada Y** debe ser un número entero aleatorio en el rango de 0 a **Canvas1.Height - Mole.Height**.



El procedimiento MoveMole, lo que sitúa al topo en un lugar al azar

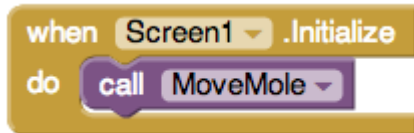
10.- Llamando al procedimiento MoveMole cuando se inicia la aplicación

Ahora que ya has programado el procedimiento **MoveMole**, vamos a hacer uso de él. Teniendo en cuenta que es muy común que l@s programador@s quieran que suceda algo cuando se inicia una aplicación, hay un bloque para ese mismo propósito:



Screen1.Initialize .

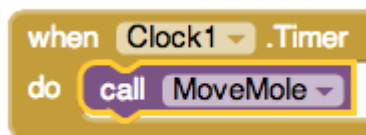
1. Haz clic en la paleta **Screen1**, y arrastra **Screen1.Initialize** .
2. Haz clic en la paleta de **Procedimientos**, donde puedes encontrar una **llamada al bloque MoveMole** . Arrástralo hacia el bloque **Screen1.Initialize**.



11.- Llamando al procedimiento MoveMole cada segundo

Hacer que el topo se mueva cada segundo requerirá un componente **Clock** (reloj) . Dejamos la propiedad **TimeInterval** de **Clock1** en su valor predeterminado de 1000 (milisegundos), o 1 segundo. Eso significa que cada segundo el bloque **Clock1.Timer** se llevará a cabo. Lo programarás así:

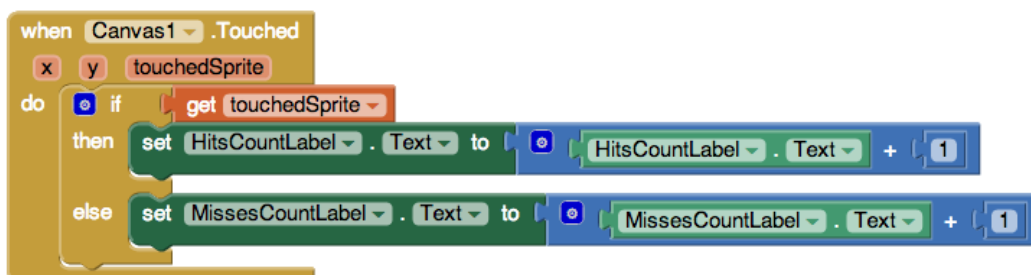
1. Haz clic en la paleta **Clock1**, y arrastra **Clock1.Timer** .
2. Haz clic en la paleta de **Procedimientos** y arrastra un bloque de **llamada MoveMole** en el **Clock1.Timer**.



Si eso es demasiado rápido o lento , puedes cambiar **la propiedad TimeInterval de Clock1** en el **Designer** para hacer que se mueva con más o menos frecuencia.

12.- Contar impactos y fallos

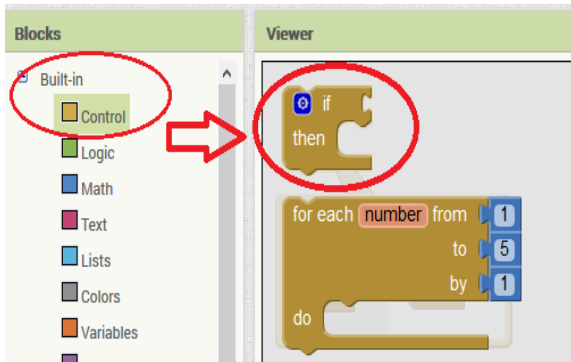
Como recordarás, se han creado dos etiquetas, **HitsCountsLabel** y **MissesCountsLabel** , que tenían valores iniciales de 0. Nos gustaría incrementar los números en estas etiquetas cada vez que el usuario toca con éxito el topo (un impacto) o toca la pantalla sin tocar el topo (un fallo). Para ello, vamos a utilizar el bloque **Canvas1.Touched** , lo que indica que el lienzo fue tocado, las **coordenadas X e Y** donde fue tocado (que no necesitamos saber), y si un sprite fue tocado (que sí necesitamos saber). Figura 9 muestra el código que va a crear.



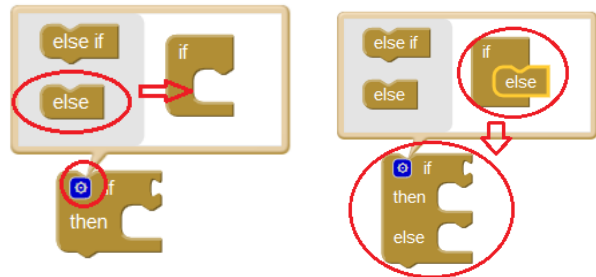
Mediante estos bloques comprobamos si dentro del **Canvas1** se tocó al **Sprite Mole** o no. Si **Mole** se toca, agrega uno al número de **HitsCountLabel.Text** , de lo contrario, agrega uno a **MissesCountLabel.Text** .

Aquí puedes ver cómo crear los bloques:

Haz clic en la paleta **Canvas1**, y arrastra un **Canvas1.Touched** .
Haz clic en la paleta de **Control** y arrastra un bloque **if then** ,

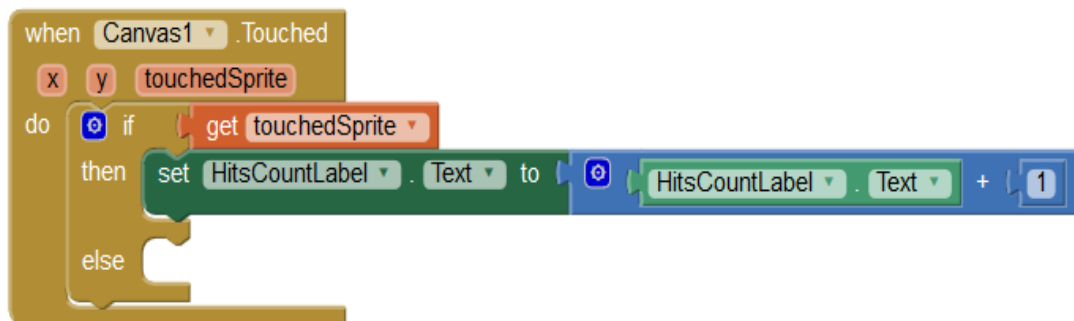


Tendrás que añadir el bloque **else** después de haberlo trasladado al área de trabajo, situándolo en **Canvas1.Touched**.



Haz clic en la paleta **Variables** y arrastra un bloque **get** y selecciona **touchedSprite** en el menú y lo colocas en el conector "if".

Como queremos que **HitsCountLabel.Text** se incremente si la respuesta a la pregunta "si" (if) es afirmativa "then", de la paleta **HitsCountLabel.Text**, arrastra el **HitsCountLabel.Text** a la derecha de una suma y a la izquierda coloca el número uno:



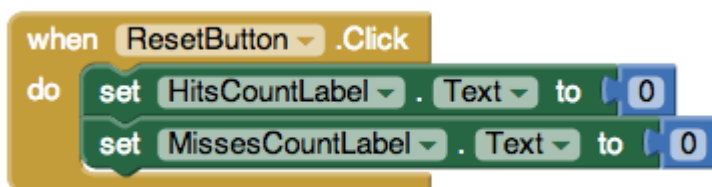
Repite las operaciones para la segunda opción "else", pero para los errores :



Prueba tu aplicación. Puedes probar este nuevo código, toca el lienzo con topo y sin topo y observa las puntuaciones.

13.- Restablecimiento de la puntuación

Para poder empezar otra vez el juego, necesitamos poner a cero los marcadores de puntuación. Lo haremos con el bloque del botón de **Reset**. Haremos que un bloque de **ResetButton.Click** establezca los valores de **HitsCountLabel.Text** y **MissesCountLabel.Text** a **0**. Crea los bloques mostrados en la **Figura 10**:

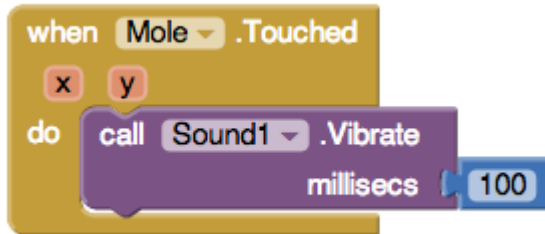


Prueba la aplicación. Trata de golpear y fallar y luego pulsa el botón Reset.



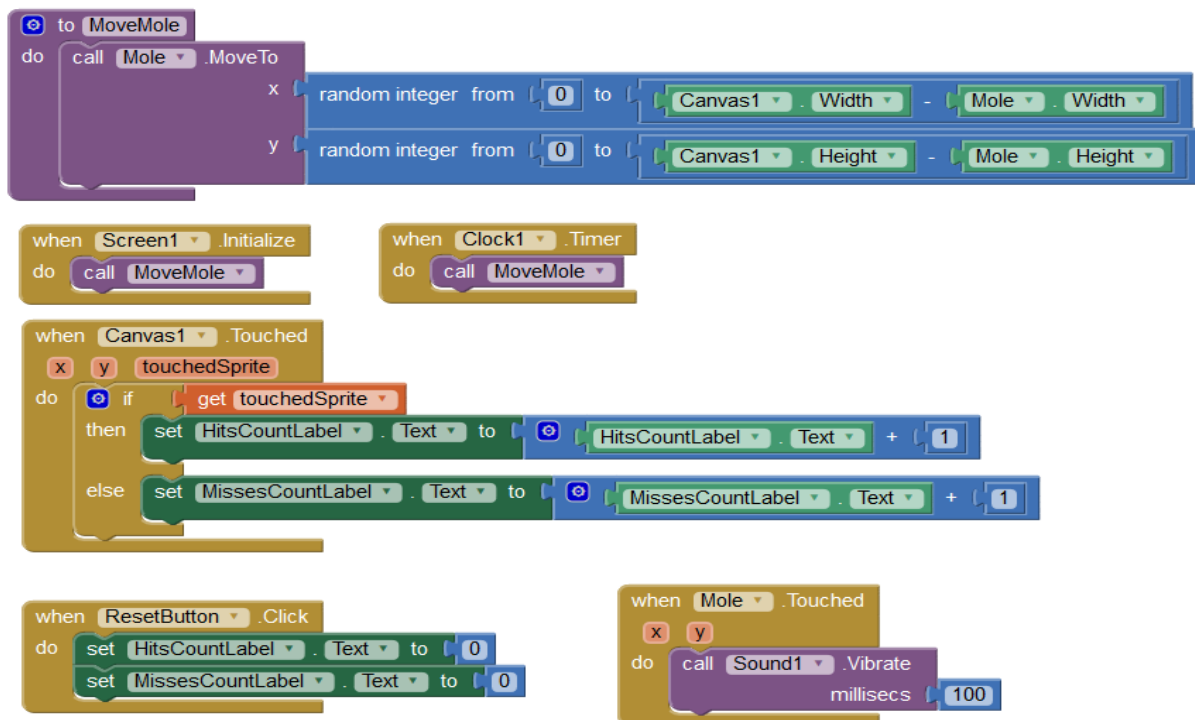
14.- Agregar comportamiento cuando se golpea al topo

Hemos dicho anteriormente que queremos que el dispositivo vibre cuando se toca el topo, podemos hacerlo con el bloque **Sound1.Vibrate** , como se muestra en la **Figura 11**:



Prueba la aplicación. Comprueba el funcionamiento completo de la aplicación, la vibración se produce cuando se golpea al topo. Si la vibración es demasiado larga o demasiado corta para tu gusto, cambia el número de milisegundos en el bloque Sound1.Vibrate .

La aplicación MoleMash completa:



Variaciones

Aquí están algunas ideas para añadir a MoleMash:

- Agrega botones para que el usuario haga que el topo se mueva más rápido o más lento.
- Añadir una segunda **ImageSprite** con un dibujo de algo que el usuario no deba golpear, como una flor. Si el usuario lo toca, le penalizará reduciendo su puntuación o terminar el juego.
- En lugar de utilizar una imagen de un topo, que el usuario pueda seleccionar una foto con el **ContactPicker** componente.

Debes de realizar una de estas variaciones o añadir otro tipo de comportamiento que tú mismo diseñes.