

## MoleMash

### Que estás construyendo

En el juego MoleMash, un topo aparece en posiciones aleatorias en un campo de juego, y el jugador consigue puntos golpeando al topo antes de que desaparezca. Este tutorial muestra cómo construir MoleMash como un ejemplo de juego simple que usa una animación.

En este tutorial se supone que has completado alguno de los tutoriales [HelloPurr](#) o [PaintPot](#).

### Primeros pasos

Conéctate al sitio web de [App Inventor 2](#) y comienza un nuevo proyecto. Llámalo **MoleMash**, y también establece el **Título** de la pantalla **Screen1** a **MoleMash**. Abre el Editor de bloques y conecta el teléfono.

También descarga esta [imagen de un topo](#) y guárdala en tu ordenador.

### Introducción

Diseñarás el juego para que el topo se mueva cada medio segundo. Si es tocado la puntuación se incrementará en un punto y el teléfono vibrará. Pulsando restart pondrás el contador a cero.

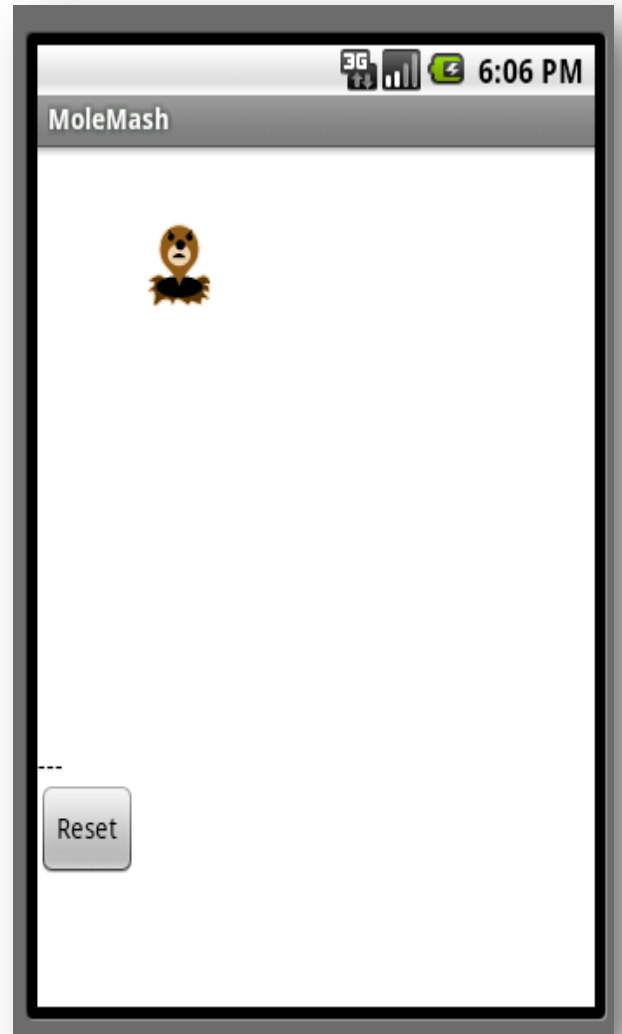
Este tutorial introduce:

- Sprites de imagen
- Temporizadores y el componente **Reloj**
- Procedimientos
- Tomar un número aleatorio entre 0 y 1
- Bloques de texto
- Typeblocking (creación de bloque mediante el teclado)

### Los primeros componentes

Muchos componentes te resultarán familiares de tutoriales anteriores:

- Un **Lienzo** llamado **MyCanvas**. Esta es el área donde el topo se moverá.
- Una **Etiqueta** llamada **ScoreLabel** que muestra la puntuación, p. ej. el número de veces que el jugador golpea el topo.
- Un **Botón** llamado **ResetButton**



Arrastra estos componentes desde la **Paleta** al **Visor** y asígnales sus nombres. Colócalo en **MyCanvas** arriba y ajusta sus dimensiones a 300 píxeles de ancho por 300 píxeles de alto. Cambia la propiedad **Texto** de **ScoreLabel** a **Score: ---**. Cambia la propiedad **Texto** de **ResetButton** a **Reset**. Añade también un componente **Sonido** y nómbralo **Noise**. Usarás **Noise** para hacer que el teléfono vibre cuando el topo sea tocado, del mismo modo que hiciste con el ronroneo del gato en **HelloPurr**.

### Temporizadores y el componente **Reloj**

---

Necesitarás que el topo salte periódicamente, y harás esto con la ayuda de un componente **Reloj**.

El componente **Reloj** proporciona varias operaciones relacionadas con el tiempo, como decirte la fecha. Aquí, usarás el componente como un temporizador que se dispare a intervalos regulares.

El intervalo de disparo está determinado por la propiedad **IntervaloDelTemporizador** de **Reloj**. Arrastra un componente **Reloj**; se colocará en el área de **Componentes no visibles**. Nómbralo **MoleTimer**. Ajusta su **IntervaloDelTemporizador** a 500 milisegundos para hacer que el topo se mueva cada medio segundo. Asegúrate de que **TemporizadorHabilitado** está marcado.

### Añadiendo un **SpriteImagen** (imagen animada)

---

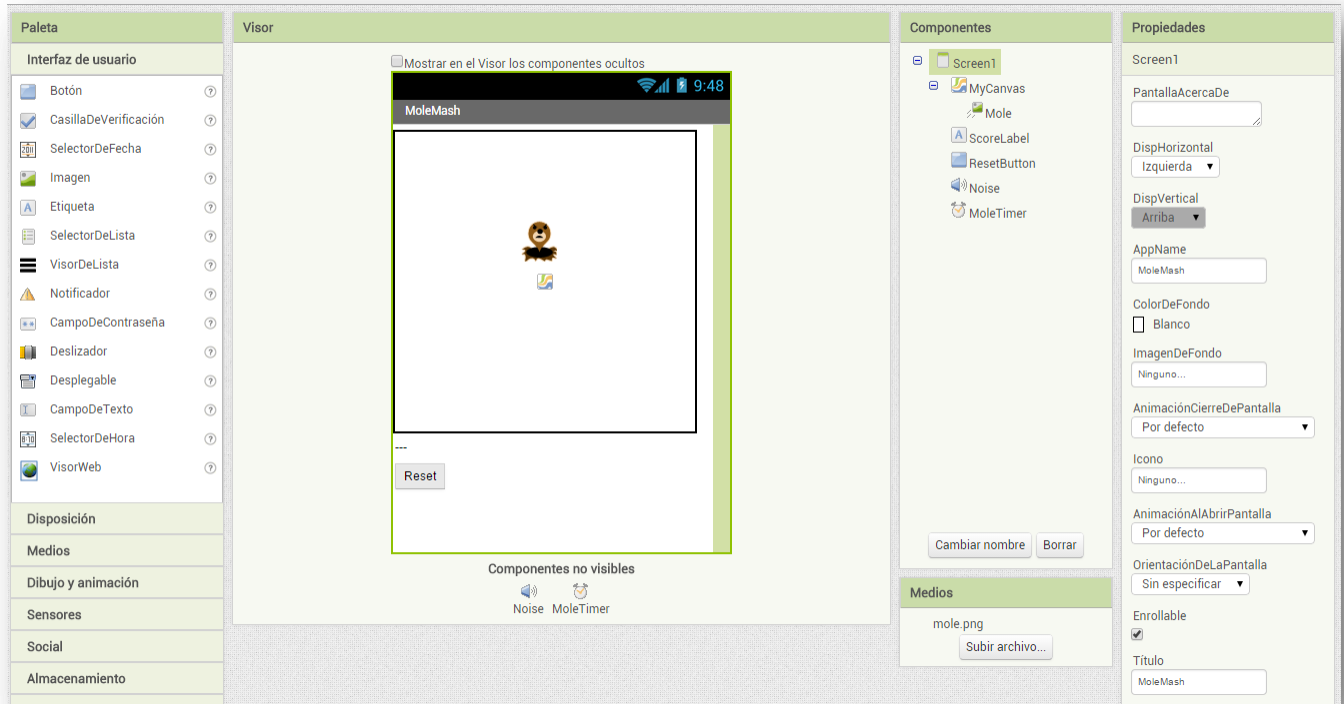
Para agregar el topo en movimiento vamos a utilizar un **SpriteImagen**.

Los **Sprites** son imágenes que se mueven en la pantalla en un **Lienzo**. Cada **Sprite** tiene una velocidad y una dirección, y también un intervalo que determina la frecuencia de los movimientos del **Sprite** a su velocidad señalada. Los **Sprites** también pueden detectar cuando se tocan. En **MoleMash**, el topo tiene una velocidad cero, por lo que no se mueve por sí mismo. En su lugar, vas a configurar la posición del topo cada vez que se active el temporizador. Arrastra un componente **SpriteImagen** al **Visor**. Encontrarás este componente en la categoría **Dibujo y animación** de la **Paleta**. Colócalo dentro del área **MyCanvas**. Establece estas propiedades para el sprite **Mole**:

- **Foto**: Usa mole.png, el cual has descargado al inicio de este tutorial.
- **Habilitado**: marcado
- **Intervalo**: 500
- El intervalo no importa aquí, porque la velocidad del topo es cero: no se mueve por sí mismo.
- **Dirección**: 0 La dirección no importa aquí tampoco, porque la velocidad es 0.
- **Velocidad**: 0.0
- **Visible**: activado
- **Ancho**: Automático
- **Alto**: Automático

Debes ver la propiedades **X** e **Y** ya rellenas. Estas fueron determinadas por el lugar donde colocaste el **SpriteImagen** cuando lo arrastraste a **MyCanvas**. Arrastra un poco más al topo. Debes ver cómo cambian **X** e **Y**. Debes ver también al topo en tu teléfono conectado, y moverse por el teléfono cuando lo arrastras en el Diseñador. Ya has especificado todos los componentes. El **Diseñador** debería parecerse a esto. Fíjate en que **Mole** está incluido bajo **MyCanvas** en la estructura de la lista de componentes, indicando que el **SpriteImagen** es

un sub-componente del **Lienzo**.



## Comportamiento de componentes y controladores de eventos

Ahora vamos a especificar el comportamiento de los componentes. Esto introduce algunas ideas nuevas de App Inventor. La primera es la idea de un **procedimiento**.

Un procedimiento es una secuencia de instrucciones a las que se puede hacer referencia a todas a la vez con un único comando. Si tienes una secuencia que es necesario utilizar más de una vez en un programa, esta se puede definir como un procedimiento, para no tenerla que repetir cada vez que la uses. Los procedimientos en App Inventor pueden tomar argumentos y devolver valores. Este tutorial cubre sólo el caso más simple: los procedimientos que no tienen argumentos y no devuelven ningún valor.

### Definir los procedimientos

Define dos procedimientos:

- **MoveMole** mueve el **SpriteImagen** Mole a una nueva posición al azar en el lienzo.
- **UpdateScore** muestra la puntuación, al cambiar el texto de **ScoreLabel**

Comienza con **MoveMole**:

- En el Editor de bloques, bajo **Integrados**, abre el cajón **Procedimientos**. Arrastra un bloque **como procedimiento** y cambia la etiqueta **procedimiento** a **MoveMole**.

**Nota:** Hay dos bloques similares: **procedimiento** y **procedureConResultado**. Aquí debes utilizar **procedimiento**.

El bloque **como MoveMole** tiene una ranura con la etiqueta **ejecutar**. Ahí es donde debes poner las declaraciones para el procedimiento. En este caso habrá dos estados: uno para establecer la posición del topo **x** y otro para establecer su posición **y**. En cada caso, establecerás la posición a una fracción aleatoria, entre 0 y 1, de la diferencia entre el tamaño del lienzo y el tamaño del topo. Crearás el valor usando el bloque **entero aleatorio** y multiplicación y resta. Encontrarás estos bloques en el cajón **Matemáticas**.

Construye el procedimiento **MoveMole**. La definición completa debe tener este aspecto:



- Observa cómo los bloques se conectan entre sí: La primera instrucción utiliza el bloque **Mole.X como** para definir la posición horizontal de topo. El valor conectado a la ranura del bloque es el resultado de multiplicar:
  - el resultado del bloque **decimal aleatorio**, el cual tiene un valor entre 0 y 1
  - el resultado de restar la anchura del topo de la anchura del lienzo.

La posición vertical se maneja de manera similar.

Con **MoveMole** hecho, el siguiente paso es definir una variable para la puntuación llamada **score** para mantener la puntuación, (número de aciertos) y darle valor inicial 0. También tenemos que definir un procedimiento **UpdateScore** que muestre la puntuación en **ScoreLabel**. El contenido actual para mostrar en **ScoreLabel** es el texto "Score: " tomado del valor de **score**.

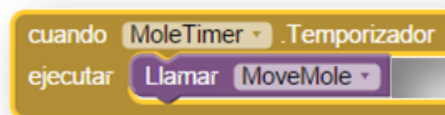
- Para crear la parte de la etiqueta "Score: ", arrastra un bloque **texto** desde el cajón **Texto**. Escribe el bloque a "Score: " en el hueco " ".
- Utiliza un bloque **unir** para unir este al bloque que da el valor de la variable **score**. Puedes encontrar el bloque **unir** en el cajón **Texto**.
- Así es como **score** y **UpdateScore** deben aparecer:



### Añade un temporizador

El siguiente paso es hacer al topo mantenerse en movimiento. Aquí es donde vamos a usar **MoleTimer**. El componente **Reloj** tiene un controlador de eventos llamado **cuando...Timer** que se dispara repetidamente a una velocidad determinada por **TimerInterval**.

Establece **MoleTimer** para llamar a **MoveMole** cada vez que se active el temporizador, mediante la construcción de un controlador de eventos como este:



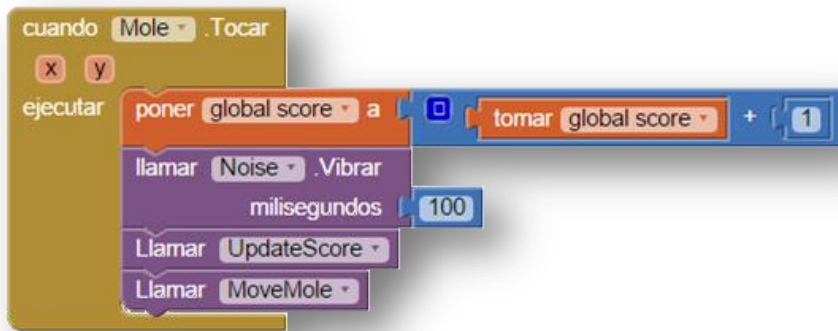
Observa cómo el topo empieza a moverse en el teléfono tan pronto como se define el controlador de eventos. Este es un ejemplo de cómo las cosas en App Inventor comienzan a suceder de forma instantánea, tan pronto como se definen.

### Agrega un controlador táctil para el topo

El programa debe incrementar la puntuación cada vez que se toque el topo. Los sprites, como los lienzos, responden a los eventos táctiles. Así que crea un controlador de eventos táctil para **Mole** que:

1. Incremente la puntuación.
2. Llame a **UpdateScore** para mostrar la nueva puntuación.
3. Hacer que el teléfono vibre durante 1/10 de segundo (100 milisegundos).
4. Llame a **MoveMole** para que el topo se mueva de inmediato, en lugar de esperar al temporizador.

Así es como debe aparecer esto en bloques. Sigue y ensambla los bloques **Mole.Touched** como se muestran.



He aquí un consejo: Puedes utilizar el **teclado** para crear rápidamente los bloques, esto se denomina **Typeblocking**.

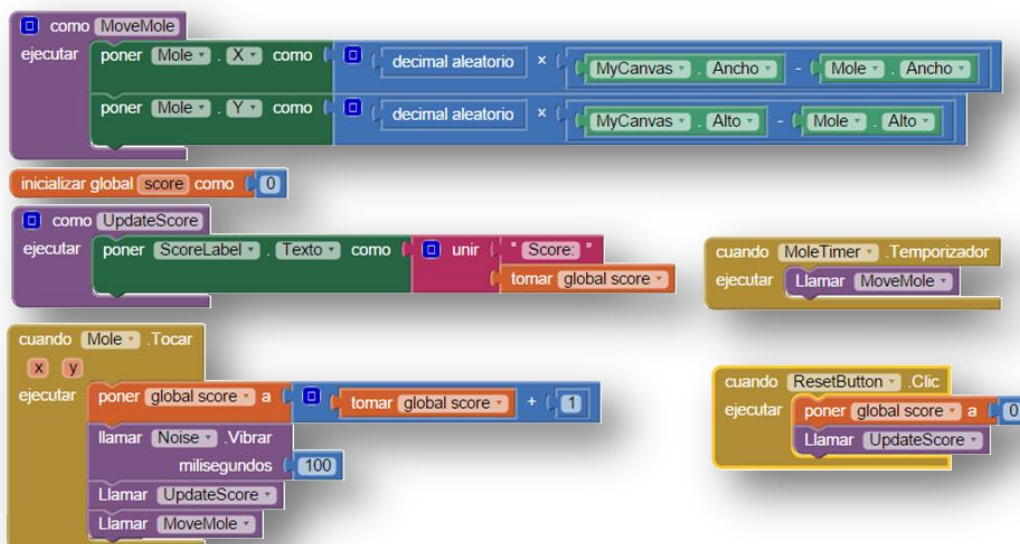
- Para crear un bloque de valor 100, sólo tienes que escribir con el teclado **100** y pulsar Intro.
- Para crear un bloque **MoveMole**, sólo tienes que escribir con el teclado **MoveMole** seleccionar el bloque que desees de la lista que aparece

### Restablecer la puntuación

Un detalle final es restablecer la puntuación. Esto simplemente es cuestión de hacer que el botón **Reset** cambie la puntuación a 0 y llame a **UpdateScore**.

### Programa Completo

Aquí está el programa MoleMash completo:



Ir a la [Página original](#) y consultar también

