

Bote de pintura (Parte 1)

Este tutorial tiene dos partes: [Parte 1](#) | [Parte 2](#)

Este tutorial presenta el componente **Canvas** para crear gráficos bidimensionales simples. Crearás una aplicación que te permitirá dibujar en la pantalla del teléfono en diferentes colores.

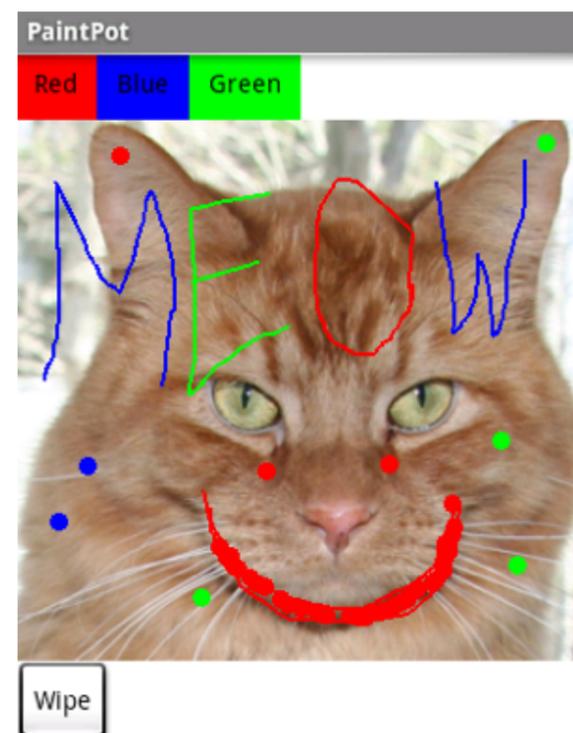
Nota histórica : PaintPot fue uno de los primeros programas desarrollados para demostrar el potencial de los ordenadores personales, ya en los años 1970. [Descargar Capítulo del Libro PDF](#)

Con la aplicación *PaintPot* , puedes:

- Sumerge tu dedo en un bote de pintura virtual para dibujar ese color.
- Arrastre el dedo por la pantalla para dibujar una línea.
- Toque la pantalla para hacer puntos.
- Utilice el botón en la parte inferior para limpiar la pantalla.
- Incluye una imagen como fondo de dibujo.

Este tutorial asume que ha completado el tutorial [de HelloPurr](#) . Este tutorial presenta los siguientes conceptos de App Inventor:

- El componente **Canvas** para dibujar.
- Controlar el diseño de la pantalla con los componentes del **Arrangement** .
- Controladores de eventos que toman argumentos.
- Variables.



Antes de empezar

Asegúrese de que su computadora y su dispositivo móvil estén [configurados para usar App Inventor](#) . Abra App Inventor en una ventana del navegador, inicie un nuevo proyecto y asígnele el nombre "*PaintPot*" .

Título de la pantalla

Para comenzar, vaya al panel Propiedades a la derecha de la ventana del Diseñador y cambie el **título** de la pantalla a "PaintPot". Si se conecta a Companion en un dispositivo móvil, debería ver este cambio en el dispositivo, con el nuevo título mostrado en la barra de título.

Hay tres nombres en App Inventor y es fácil confundirlos:

1. El nombre que eliges para tu proyecto mientras trabajas en él (en este caso, *PaintPot*). Este también será el nombre de la aplicación si la empaquetas para el teléfono.
2. El nombre "Pantalla1", que es el nombre del componente **Pantalla** . Lo verá listado en el panel Componentes del Diseñador. No puede cambiar el nombre del primer componente de pantalla en la versión actual de App Inventor, pero puede crear pantallas adicionales con los nombres de cualquier cosa que deba elegir.
3. La propiedad **Título** de la pantalla, que es lo que verá en la barra de título del dispositivo móvil. El título es una propiedad del componente **Pantalla** . El título comienza siendo "Pantalla1", que es lo que usaste en *HelloPurr* . Sin embargo, puedes cambiarlo, como lo estás haciendo con *PaintPot* . Para reiterar, el nombre y el título de Pantalla1 son inicialmente los mismos, pero puedes cambiar el título si lo deseas.

Configurar los componentes

Utilizarás estos componentes para crear *PaintPot* :

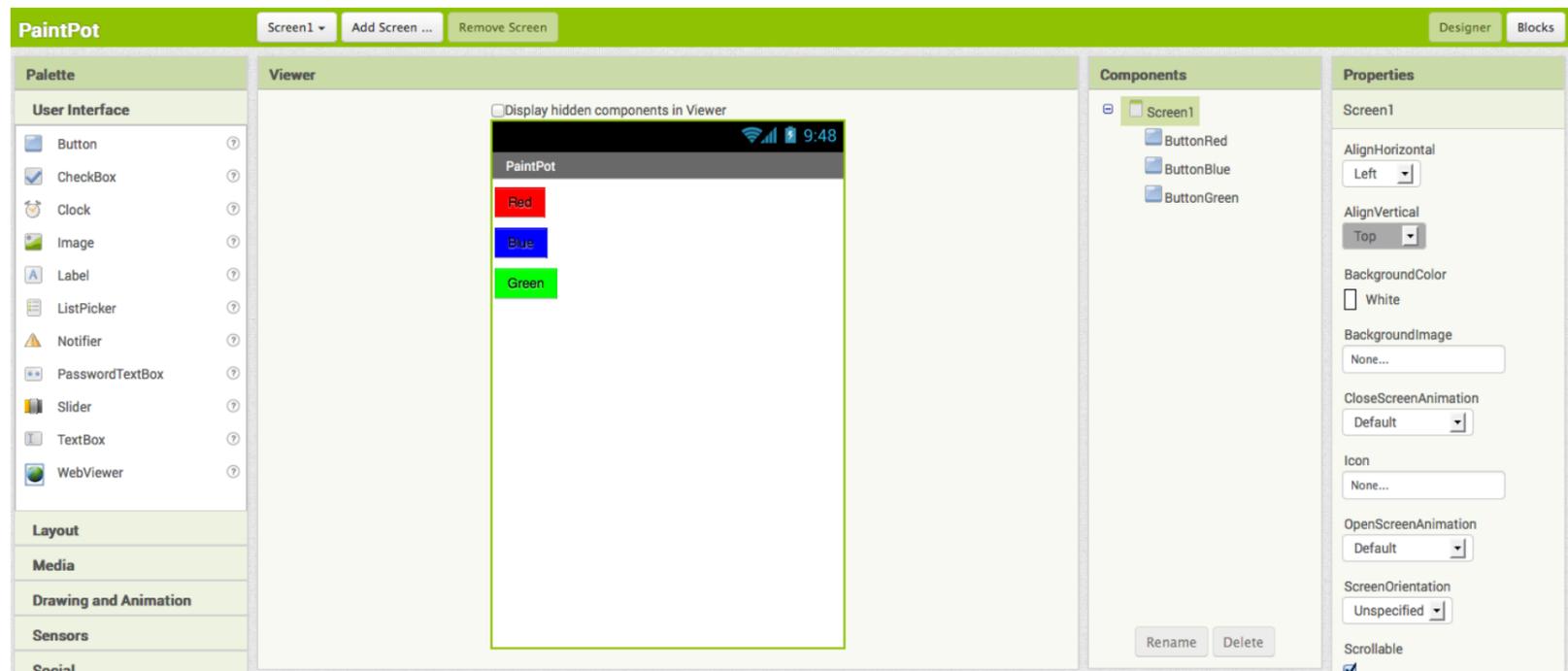
- Tres **botones** para seleccionar pintura **roja** , **azul** o **verde** y otro botón para borrar el dibujo.
- Un **Canvas** , la superficie de dibujo. Este lienzo tiene una **BackgroundImage** , que es [este gatito](#) del tutorial *HelloPurr* . También puedes dibujar sobre un lienzo en blanco. Eso es sólo un lienzo sin imagen de fondo.
- También hay un componente que no ves: usas un **HorizontalArrangement** para hacer que los tres botones de colores se alineen uno al lado del otro.

Eso hace cinco componentes en total. Consiguámoslos y creemos la aplicación.

Botones de colores

- Arrastre un componente **de botón** al visor y cambie el atributo de *texto* del botón a "Rojo" y haga que su *color de fondo* sea rojo.
- Haga clic en **el Botón1** en la lista de Componentes del Visor para resaltarlo (es posible que ya esté resaltado) y use el botón **Cambiar nombre...** para cambiar su nombre de "Botón1" a "BotónRojo".
- De manera similar, haz dos botones más para el azul y el verde, llamados "ButtonBlue" y "ButtonGreen", colocándolos verticalmente debajo del botón rojo.

Así es como debería verse esto en el Diseñador, con los nombres de los botones apareciendo en la lista de componentes del proyecto. En este proyecto, cambiará los nombres de los componentes en lugar de dejarlos con los nombres predeterminados como lo hizo con *HelloPurr* . El uso de nombres significativos hace que sus proyectos sean más legibles para usted y los demás.



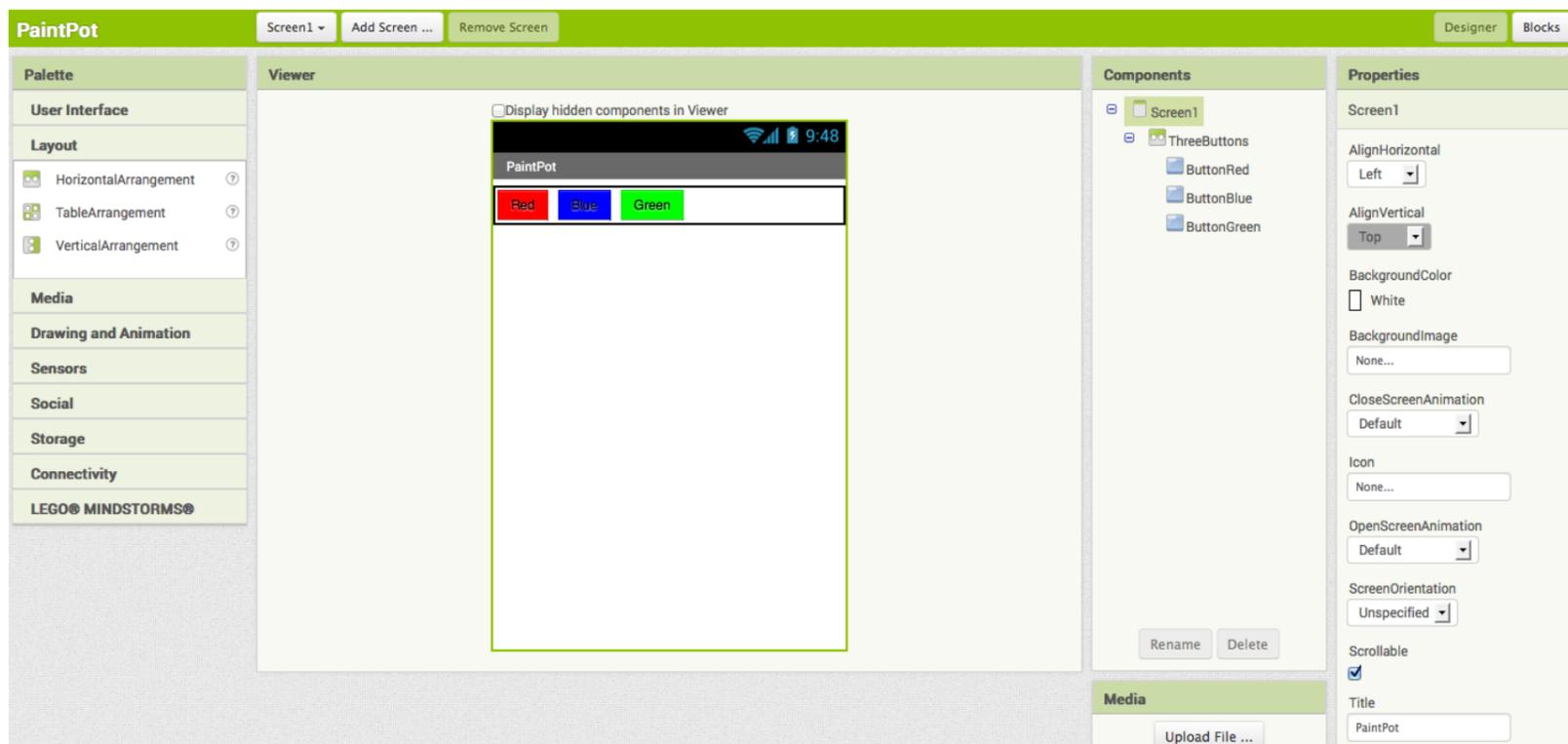
Conéctese al AI2 Companion usando el menú "Conectar". También deberías ver los tres botones en la pantalla del dispositivo móvil.

Diseño con disposición de pantalla

Ahora deberías tener tres botones, uno debajo del otro. El siguiente paso es alinearlos horizontalmente. Esto se hace utilizando un componente **HorizontalArrangement** .

1. Desde el cajón Diseño de la paleta, arrastre un componente **HorizontalArrangement** y colóquelo debajo de los botones. Cambie el nombre de este componente de "HorizontalArrangement1" a "ThreeButtons".
2. En el panel Propiedades, cambie el *ancho* de **ThreeButtons** a "Rellenar padre..." para que ocupe todo el ancho de la pantalla.
3. Arrastre los tres botones uno al lado del otro al componente **HorizontalArrangement** . **Sugerencia** : Verás una línea vertical azul que muestra dónde irá la pieza que estás arrastrando.

Si observa la lista de componentes del proyecto, verá los tres botones sangrados debajo de **ThreeButtons** para mostrar que ahora son sus subcomponentes. Observe que todos los componentes tienen sangría en **Pantalla1** .



También debería ver sus tres botones alineados en una fila en la pantalla de su dispositivo móvil, aunque es posible que las cosas no se vean exactamente como en el Diseñador. Por ejemplo, el esquema del arreglo se muestra en el Diseñador pero no en el dispositivo.

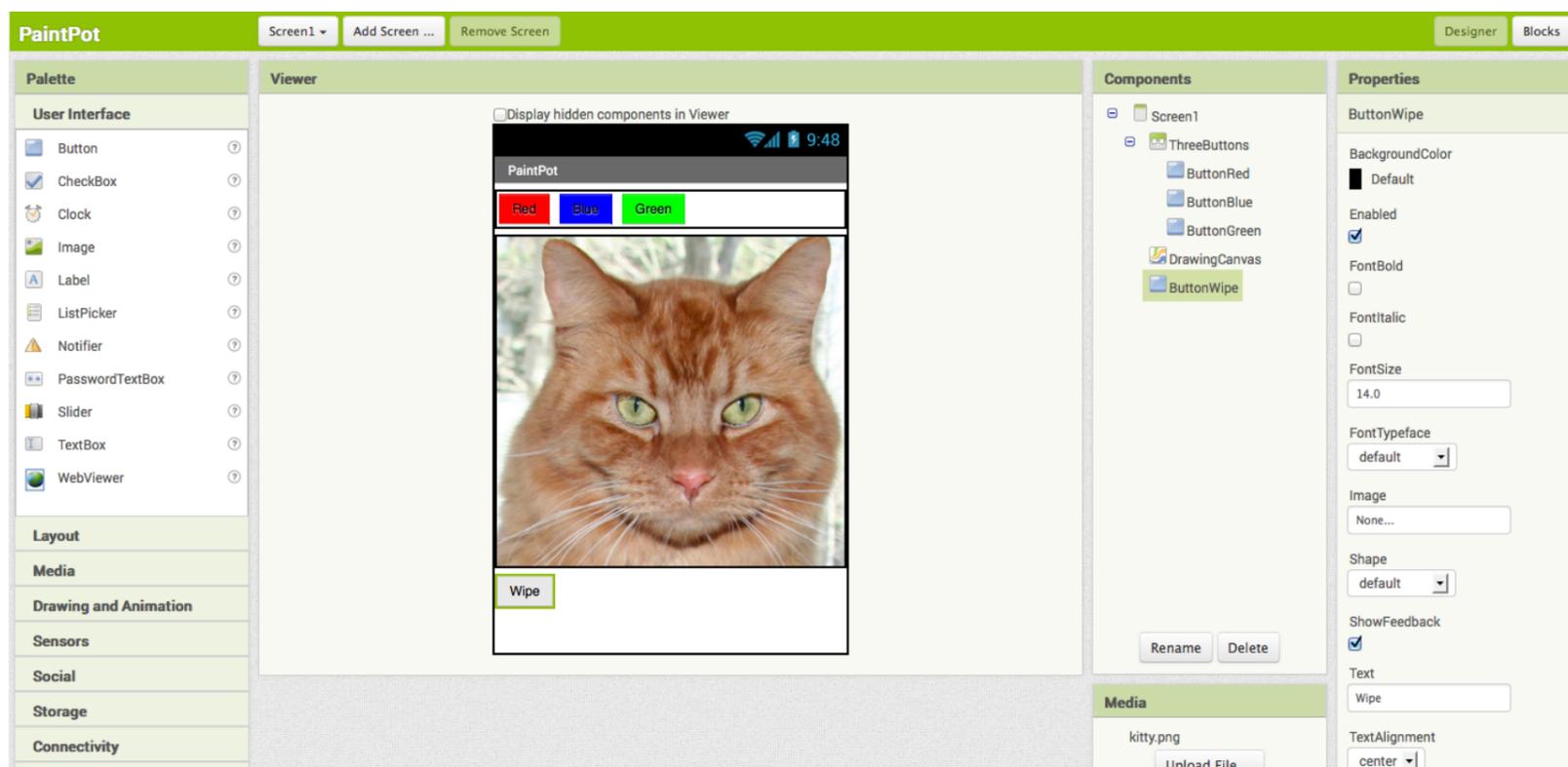
En general, se utilizan Disposiciones de diseño para crear diseños verticales u horizontales simples. Puede crear diseños más complejos anidando componentes de Disposición de pantalla. También hay un componente **TableArrangement** (no cubierto en este tutorial).

Botón de lienzo y limpieza

Los dos últimos componentes son el lienzo y el botón de limpieza.

1. Desde el cajón de Dibujo y Animación de la Paleta, arrastre un componente **de Lienzo** al Visor. Cambie su nombre a "DrawingCanvas". Establezca su *Ancho* en "Rellenar padre" y establezca su *Alto* en 300 píxeles.
2. Agregue una imagen de fondo al **lienzo**. Haga clic en el campo que contiene "Ninguno..." junto a *Imagen de fondo* en el panel Propiedades del lienzo. Puedes usar el mismo archivo [kitty.png](#), del tutorial de HelloPurr, o puedes usar otra imagen de tu elección.
Puede usar cualquier imagen que desee, pero obtendrá los mejores resultados si el tamaño de la imagen (en píxeles) es cercano al tamaño en el que la mostrará en el dispositivo. Además, las imágenes grandes tardarán mucho en cargarse y pueden exceder la capacidad de memoria que el dispositivo asigna para las aplicaciones.
3. Desde la Paleta, arrastre el último **Botón** a la pantalla y colóquelo debajo del Lienzo. Cambie el nombre de este componente de botón a "ButtonWipe" y cambie su propiedad *Texto* a "Wipe".

Ahora ha completado los pasos para configurar la apariencia de su aplicación. Así es como debería verse esto en el Diseñador. A continuación, definirá cómo se comportan los componentes.



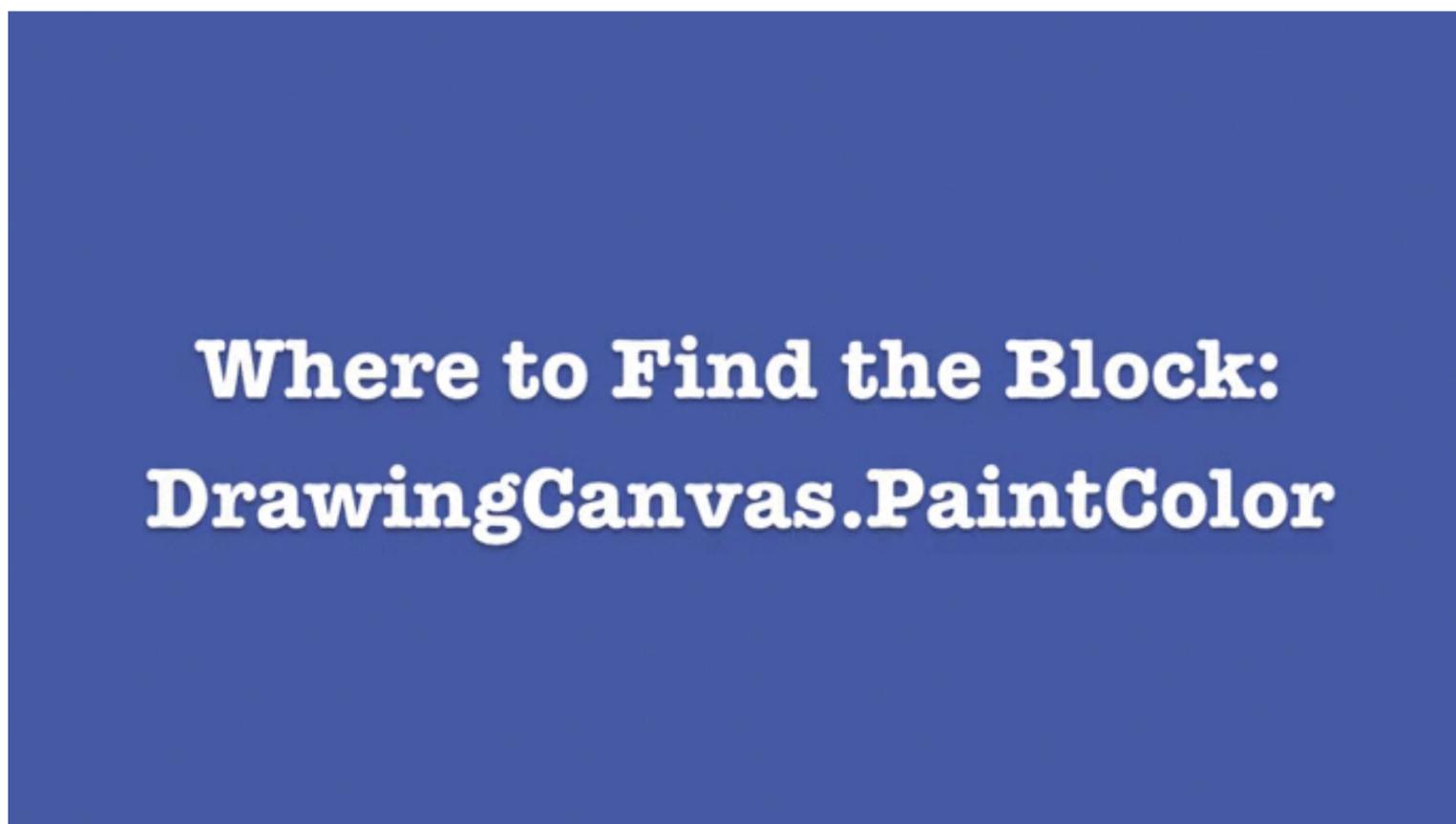
Agregar comportamientos a los componentes

Haga clic en el botón Bloques para cambiar al Editor de bloques. Primero codificarás los botones que cambian el color de la pintura. Luego agregarás bloques para codificar lo que sucede cuando alguien toca o arrastra la pantalla.

Agregar controladores de eventos de botón

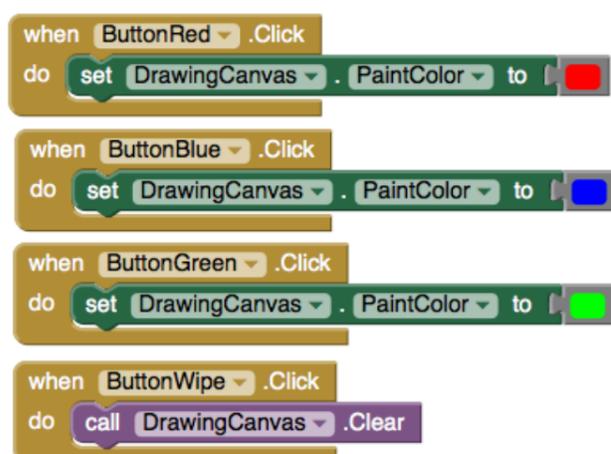
En el editor de bloques:

1. Abra el cajón de **ButtonRed** y arrastre el bloque `cuando ButtonRed.Click`.
2. Abra el cajón **DrawingCanvas**. Arrastre el conjunto `DrawingCanvas.PaintColor` para bloquear (recuerde que el bloque establecido de componentes es un [menú desplegable](#), por lo que `PaintColor` es una selección en el menú desplegable) y colóquelo en la sección `hacer` de `cuando ButtonRed.Click`.



3. Abra el cajón de Colores y arrastre el bloque del color Rojo y ajústelo para establecer `DrawingCanvas.PaintColor` en `.` (Al hacer clic en un bloque de color después de haberlo colocado, se mostrará una tabla de colores entre los que puede seleccionar).
4. Repita los pasos 2 a 4 para los botones azul y verde.
5. El último botón para configurar es el botón Limpiar. Cree un controlador de eventos de clic para **ButtonWipe** arrastrando `cuando ButtonWipe.Click` desde el cajón de **ButtonWipe**. Desde el cajón **DrawingCanvas**, arrastre la llamada `DrawingCanvas.Clear` y colóquela en el área de `hacer` del bloque `cuando ButtonWipe.Click`.

Los bloques para los botones deberían verse así:



Agregar controladores de eventos táctiles

Ahora vamos al siguiente paso: dibujar en el **lienzo**. Agregarás bloques para que cuando toques el lienzo, obtengas un punto en el lugar donde tocas. Si arrastras el dedo lentamente por el lienzo, se dibuja una línea.

- En el Editor de bloques, abra el cajón del lienzo y arrastre el bloque `when DrawingCanvas.Touched` al espacio de trabajo. Tan pronto como arrastre el bloque hacia afuera, podrá notar tres nombres de argumentos (resaltados en naranja) ubicados en la parte superior del bloque `x`, `y` y `touchSprite`. Estos argumentos también se conocen como variables locales y se puede acceder a ellos utilizando los bloques `get` o `set` que se encuentran en el cajón de Variables y luego seleccionando la variable adecuada en el menú desplegable. También puede acceder a estas variables moviendo el cursor sobre el nombre resaltado y seleccionando la variable que le gustaría usar.

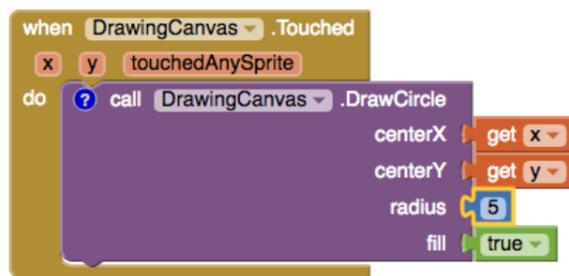
Ya has visto eventos de clic en botones. Los clics son simples, porque no hay nada que saber sobre el clic aparte de que ocurrió. Otros controladores de eventos, como `cuando... Tocado`, necesitan información sobre el evento. En App Inventor, esta información se expresa como el valor de los argumentos asociados con el controlador de eventos. Para el evento `cuando... Tocado`, los dos primeros argumentos representan las coordenadas xey del lugar donde ocurrió el toque. Guardaremos **TouchSprite** para un tutorial posterior.

- Para este evento táctil, haga que el lienzo dibuje un pequeño círculo en el punto con coordenadas (x, y). Arrastre un bloque `llamado DrawingCanvas.DrawCircle` desde el cajón de Canvas y colóquelo en la sección `hacer de cuando DrawingCanvas.Touched`.

En el lado derecho del bloque `llamado DrawingCanvas.DrawCircle` hay cuatro sockets donde debe especificar valores para las coordenadas x e y donde se debe dibujar el círculo, **radio**, que es el radio del círculo, y **relleno**, que es verdadero para dibujar un círculo relleno o falso para dibujar un círculo delineado. Para **x** e **y**, usará los valores de los argumentos que se proporcionaron al controlador Touched:

1. Mueva el cursor sobre la variable **x** (resaltada en naranja). Busque el bloque `get x` y arrástrelo al conector x correspondiente en el bloque `when DrawingCanvas.Touched`.
2. Haga lo mismo con la variable **y**.
3. También necesitarás especificar el radio del círculo a dibujar. Cinco (píxeles) es un buen valor para esta aplicación. Haga clic en un área en blanco de la pantalla y escriba el número 5 seguido de regresar para crear un bloque numérico con un valor de 5. Escribir en el área en blanco de la pantalla se llama bloqueo de tipos y es un atajo útil que debe [conocer](#). Esto se puede hacer para cualquier bloque, no solo para números. Inserte el bloque para 5 en la ranura del radio. Haga clic en los siguientes enlaces para obtener más información sobre [argumentos](#) y [bloqueo de tipos](#).
4. Arrastre un bloque verdadero del cajón Logic y conéctelo a la ranura de relleno. Esto garantiza que los puntos dibujados se completarán.

Así es como debería verse el controlador de eventos táctiles:



Intente probar lo que tiene hasta ahora en el dispositivo. Toca un botón de color. Ahora toca el lienzo y tu dedo debería dejar un punto en cada lugar que toques. Al tocar el botón Limpiar, se borrará el dibujo.

Agregar eventos de arrastre

Finalmente, agregue el controlador de eventos de arrastre. Aquí está la diferencia entre tocar y arrastrar:

- Un toque es cuando colocas el dedo sobre el lienzo y lo levantas sin moverlo.
- Un arrastre es cuando colocas tu dedo sobre el lienzo y lo mueves mientras lo mantienes en contacto.

Cuando arrastra el dedo por la pantalla, parece dibujar una línea curva gigante donde movió el dedo. Lo que en realidad estás haciendo es dibujar cientos de pequeñas líneas rectas: cada vez que mueves el dedo, aunque sea un poquito, extiendes la línea desde la última posición inmediata del dedo hasta su nueva posición.

Un evento de arrastre viene con 6 argumentos. Estos son tres pares de coordenadas xey que muestran:

- La posición de su dedo hacia atrás donde comenzó el arrastre.
- La posición actual de su dedo.
- La posición inmediatamente anterior de su dedo.

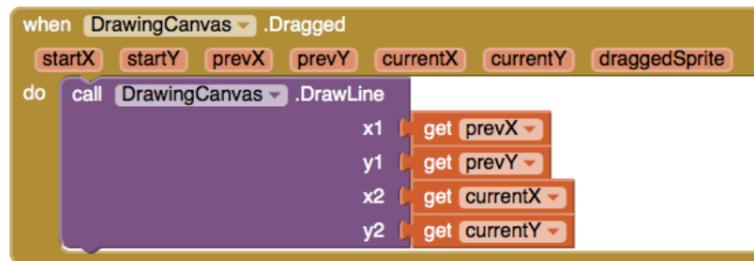
También hay un objeto, que ignoraremos en este tutorial.

Ahora haga que la aplicación dibuje una línea entre la posición anterior y la posición actual creando un controlador de arrastre:

1. Desde el cajón **DrawingCanvas**, arrastre el bloque `cuando DrawingCanvas.Dragged` al espacio de trabajo.
2. También desde el cajón **DrawingCanvas**, arrastre el bloque `de llamada DrawingCanvas.DrawLine` a la ranura `do` del bloque `when DrawingCanvas.Dragged`.

3. Arrastre un bloque `get prevX` a la ranura **x1 abierta** `cuando DrawingCanvas.DrawLine`. Haga lo mismo con los otros espacios: **y1** debe ser `get prevY`, **x2** debe ser `get currentX` y **y2** debe ser `get currentY`.

Aquí está el resultado:



Pon a prueba tu trabajo probándolo en el dispositivo: arrastra el dedo por la pantalla para dibujar líneas y curvas. Toca la pantalla para hacer puntos. Utilice el botón Limpiar para borrar la pantalla.

En [PaintPot Part 2](#), verás cómo usar variables globales para crear puntos de diferentes tamaños.

Revisar

Estas son algunas de las ideas cubiertas en este tutorial:

- Puede utilizar los componentes de Disposición de diseño para especificar diseños de pantalla además de simplemente colocar componentes uno debajo del otro.
- El componente **Canvas** te permite dibujar sobre él. También puede sentir toques y arrastres.
- A algunos controladores de eventos se les llama con información sobre el evento, como las coordenadas del lugar donde se tocó la pantalla. Esta información está representada por argumentos. Cuando selecciona un controlador de eventos que tiene argumentos, App Inventor crea bloques `de argumentos` para estos.

Escanee la aplicación de muestra en su dispositivo

Escanee el siguiente código QR en su dispositivo para instalar y ejecutar la aplicación de muestra.

Inventor de la aplicación del MIT



© 2012-2022 Instituto de Tecnología de Massachusetts Este trabajo tiene una [licencia internacional Creative Commons Atribución-CompartirIgual 4.0](#). [Termino del servicio y política de privacidad](#)

Soporte de App Inventor: [Comunidad](#)

Otras consultas: [Correo electrónico](#)

GitHub: [mit-cml](#)

Accesibilidad: [accesibilidad.mit.edu](#)

