

Lección 29: vinculación de pantallas

Lección en línea [Ver en Code Studio](#)

Propósito

En esta Lección, los equipos combinan las pantallas que diseñaron en la Lección anterior en una sola aplicación, que luego pueden vincular utilizando código. Los estudiantes aprenden programación basada en eventos básicos mediante la construcción de la aplicación modelo que comenzaron en la Lección anterior. Además de la pantalla que los estudiantes diseñaron, aprenderán a crear pantallas adicionales e incluso a importar pantallas hechas por otros.

Esta Lección y la siguiente se centran en conceptos de programación, y la medida en que lo hacen es mínima. El objetivo de esta Lección no es enseñar programación en App Lab, sino más bien enseñar la cantidad mínima de habilidades de programación para permitir que los estudiantes hagan sus aplicaciones interactivas.

Secuencia para el aprendizaje

- Conocimiento inicial (5 min)
- Ampliación del conocimiento (45-75 min)
- Transferencia del conocimiento (5 min)

Objetivos

Los estudiantes serán capaces de:

- Escribir programas que respondan a la entrada del usuario.
- Integrar pantallas diseñadas por otros en una aplicación propia.
- Colaborar con otros para desarrollar un prototipo interactivo.

Código

- `onEvent(id, type, function(event)){ ... }`
- `setScreen(screenId)`
- `console.log`

Estrategia de aprendizaje

Conocimiento inicial (5 min)

Reaccionando a eventos

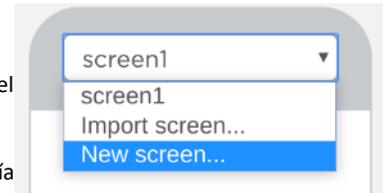
Inducción:

Recordar cuando estabas programando en Game Lab. Cuando querían responder a la interacción del usuario, ¿cómo lo hicieron?

Observaciones:

En Game Lab usamos condicionales dentro del ciclo draw() para verificar si un usuario había interactuado con el juego treinta veces por segundo. Esta técnica funciona muy bien en juegos y animaciones donde hay muchas otras cosas que también le está diciendo a la computadora que haga. La mayoría de las aplicaciones de teléfono, sin embargo, no están haciendo cosas constantemente. De hecho, muchas aplicaciones no hacen absolutamente nada más que esperar a que el usuario haga clic en algo.

Para aplicaciones como esta, es mejor usar un concepto de programación llamado “Evento”, que le permite definir qué código debe ejecutarse cuando ocurren diferentes acciones. En App Lab, los “manejadores de eventos” se pueden unir a los elementos que se crearon en el modo de diseño, donde pueden observar la ocurrencia de varias acciones (como hacer clic en un botón).



Ampliación del conocimiento (45-75 min)

Vinculación de pantallas

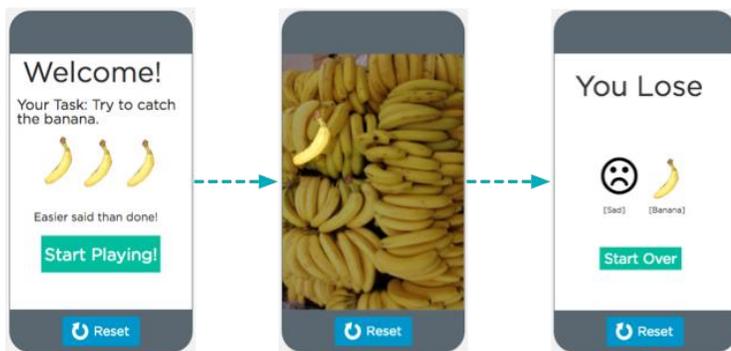
Consejo de enseñanza:

Durante esta Lección, y para el resto de esta unidad, los estudiantes deberán compartir enlaces de proyectos entre sí para importar su trabajo compartido. Considere proporcionar a cada grupo un documento compartido para rastrear los enlaces de sus proyectos a fin de facilitar este proceso.

Esta Lección comienza con la aplicación del modelo que los estudiantes comenzaron en la Lección anterior. Agregarán e importarán pantallas adicionales, y luego escribirán código que vincula sus pantallas a botones.

Niveles de Code Studio (link)

Multi-Screen Apps: Puede imaginarte que una aplicación típica tiene una cantidad de “pantallas” o vistas que atraviesa un usuario como parte del uso de tu aplicación. A continuación, se muestra un ejemplo típico con una pantalla de bienvenida, una pantalla que reproduce un juego en la pantalla



¿Qué es una pantalla?

Una pantalla es solo un elemento de la interfaz de usuario como cualquier otra cosa en el modo de diseño. Pero tiene algunos elementos especiales.

Instrucciones para el alumno

Tarjetas a pantallas

Para simplificar el proceso de creación de prototipos, los equipos pueden pensar en cada tarjeta que crearon en Paper Prototyping como una pantalla separada en la iteración del App Lab de su aplicación. En realidad, las pantallas que son muy similares a menudo se pueden desarrollar como una sola pantalla con contenido que cambia según la interacción del usuario. En la próxima Lección, los estudiantes verán algunos ejemplos de cómo podría funcionar, pero no se espera que incorporen ese tipo de funcionalidad en este proyecto.

Alcanzando

Si los estudiantes se perdieron la Lección anterior o no pudieron completar la pantalla de inicio, puede hacer que importen esta versión completa de la pantalla de inicio.

Vea el [mapa de importación de pantalla](#) para más detalles.

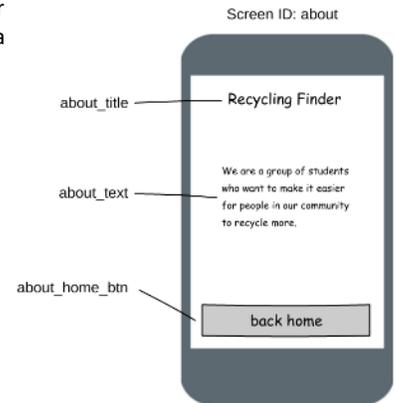
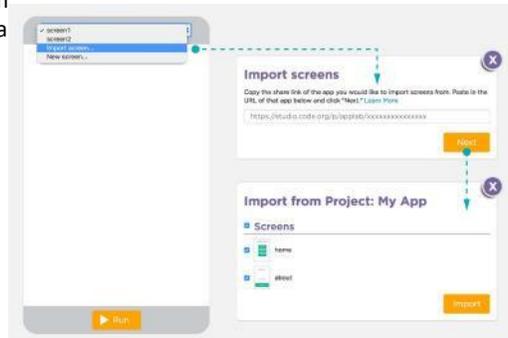
1. **Agregar una nueva pantalla:** Para completar tu aplicación, vas a necesitar más que solo una pantalla de inicio. Cada tarjeta que creaste en tu prototipo de papel está representada por una “pantalla” en App Lab. Para crear más pantallas, simplemente Haga clic en el menú desplegable en la parte superior de la pantalla de la aplicación y selecciona “Nueva pantalla ...”
2. **Diseña la pantalla Acerca de:** Para agregar elementos a su nueva pantalla Acerca de, deberá seleccionarla primero en el menú desplegable de pantallas.
3. **Importación de pantallas:** Has visto cómo crear una aplicación con varias pantallas, pero ¿qué hay de trabajar con otras personas para colaborar en una aplicación? Afortunadamente, App Lab te permite importar pantallas desde otra aplicación. Desde el mismo menú desplegable donde puede agregar una pantalla, simplemente selecciona “Importar pantalla” y sigue las instrucciones (necesitarás el

código para compartir desde la aplicación que estás importando las pantallas). Además de agregar nuevas pantallas en blanco, puedes importar pantallas desde “Importar pantalla ...”. Para importar una pantalla de otra persona, necesitarás que te proporcionen la URL para compartir.

Nota:

¡Las pantallas que importes no deben compartir ningún ID con elementos que ya estén en tu aplicación!

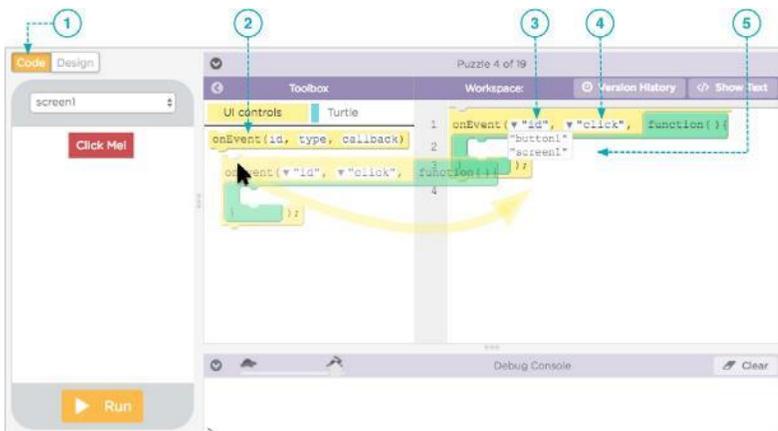
Respondiendo a la entrada del usuario



El modo de diseño hace que sea realmente fácil diseñar la interfaz de usuario para su aplicación, pero necesitaremos sumergirnos en el modo de código para que realmente responda a los usuarios. En Game Lab, el bucle Draw ejecuta código constantemente, usando condicionales para verificar las interacciones del usuario. App Lab, por otro lado, espera responder a ciertos eventos que ocurren en un programa. Es la manera en que App Lab dice “¡Oye, algo acaba de pasar!”. Esto significa que no es necesario que haya código en ejecución cuando no está interactuando directamente con una aplicación, y su código puede responder a cada uno de estos eventos a medida que ocurren.

En App Lab, todo el manejo de eventos comienza agregando un bloque especial de código llamado `onEvent()`. Aquí está cómo hacerlo:

Eventos de botones



Con nuestras pantallas en su lugar, es hora de comenzar a programar realmente la aplicación para que responda a los clics del botón. Para cada botón en cada pantalla, deseará agregar un bloque `onEvent` que “al hacer clic en ese botón, responda de manera adecuada”. Para comenzar, simplemente miraremos el botón “`home_search_btn`” e imprimiremos algo en la consola cuando se haga clic

Hacer esto

Ahora está en el modo de código (puedes usar los botones situados encima de tu aplicación para cambiar entre el modo de código y el de diseño). Aquí es donde puedes escribir el código que responderá a los usuarios que interactúan con tu aplicación.

- Arrastra un bloque `onEvent` desde la caja de herramientas del código.
- Selecciona “`home_search_btn`” del menú desplegable “`id`”.

Promoviendo el trabajo en equipo

Los siguientes niveles prepararán a los equipos para fusionar todas sus pantallas por separado en una sola aplicación. Hay dos posibles obstáculos que pueden encontrar durante el proceso de importación:

- URL mal comunicadas: Las URL compartidas son largas y complicadas, por lo que los estudiantes necesitarán una ubicación compartida donde puedan copiar/pegar sus URL
- ID de pantalla en conflicto: la importación de una pantalla con la misma ID que una existente le permitirá sobrescribir la pantalla existente. Esto es particularmente problemático cuando los estudiantes dejan la identificación de pantalla predeterminada “`screen1`”
- ID de elementos en conflicto: si los estudiantes han asignado correctamente todos sus elementos, esto no debería ser un problema, pero cualquier elemento que tenga la ID predeterminada o que comparta una ID con un elemento existente evitará que se importe una pantalla.

Una vez que los equipos pasan a la fase de programación de este proyecto, se vuelve mucho más difícil para los estudiantes colaborar en la misma aplicación. La fase de diseño es la oportunidad principal para que todos los estudiantes colaboren en la aplicación, después cada alumno trabajará en su propia copia.

Solo los hechos

Aquí hay mucho que consumir y muchos de los detalles sobre cómo funcionarán los eventos en App Lab no son totalmente necesarios para comprender nuestras necesidades en esta unidad. Si los estudiantes están luchando con la forma en que funciona la programación en App Lab, concéntrese en estos elementos esenciales:

- No hay ningún comando **draw** que se ejecute para siempre
- El bloque **onEvent ()** especifica:
 - un elemento ID
 - un evento (como "click" o "mouseover")
 - código que debe ejecutarse cuando ocurre el evento (este código es técnicamente una función de devolución de llamada anónima, representada por el bloque verde)
- Los bloques de código **Inside onEvent ()** se ejecutan en cualquier momento que ocurra el evento especificado

- Arrastra un `console.log` bloque desde el cajón de variables.
- Ejecute su aplicación.
- Haga clic en el botón Buscar y busque mensajes en la consola.

```
onEvent (▼ "home_search_btn", ▼ "click", function (event) {
  console.log ("Search clicked!");
});
```

Cambio de pantallas

Usando el comando `setScreen` para cambiar las pantallas con el código

El comando `setScreen` cambie la pantalla para mostrar la pantalla que desee. Usted especifica la pantalla por su ID.



Cambio de pantallas al hacer clic

El bloque `setScreen` cambiará tu aplicación a la pantalla que especifique. Si coloca un `setScreen` dentro de un `onEvent`, puede programar sus botones para cambiar a la pantalla correcta

Finalizando el día:

Dependiendo de la cantidad de tiempo que tenga en clase, este puede ser un buen momento para cerrar el día.

Tiempo de equipo:

Aunque los estudiantes están trabajando en sus copias individuales de la aplicación de su equipo en este punto, aliente a los equipos a ser activos durante esta parte de la Lección. Cualquier problema que los estudiantes encuentren al importar las pantallas de sus compañeros de equipo solo se puede resolver mediante el trabajo en equipo, y es probable que sea un problema para todos los miembros de este.

Comparando con Game Lab

Comparado con Game Lab

En Game Lab podríamos usar el siguiente código para verificar si se hizo clic en un sprite llamado "botón" y hacer algo al respecto:

En App Lab, dado un "botón" ID, podríamos escribir el siguiente código para lograr el mismo objetivo:

```
función draw () {
  if ( mousePressedOver ( botón ) ) {
    // Hacer algo
  }
}
```

En el ejemplo de AppLab, la función es simplemente otro argumento pasado a la función `OnEvent`, sucede que el argumento debe ser una función

```
function doSomething () {
  // Hacer algo
}

onEvent ( "botón", "click", hacer Algo );
```

en sí misma. También podríamos definir la función por separado y escribir este mismo programa de la siguiente manera:

Programación limitada requerida

El objetivo de esta unidad es hacer que los estudiantes piensen en el papel que juega el diseño en el desarrollo de un producto y ayudarlos a generar empatía con los usuarios finales. Esta no es una unidad centrada en la programación, y el grado en el que se espera que los estudiantes programen está representado en su totalidad aquí.

Combinar pantallas de proyectos

En el segundo ejemplo observa que cuando pasamos la función `doSomething` a la `onEvent` no incluimos el paréntesis al final. Esto es porque en realidad estamos pasando toda la función como un objeto.

Revisión: En la Lección anterior, cada estudiante desarrolló una pantalla para la aplicación de su equipo. Dará a los equipos unos minutos para revisar el estado de sus pantallas, identificando cualquier trabajo restante antes de combinarlos. En el último nivel de esta Lección pedirá a los estudiantes que importen y vinculen cada una de las pantallas en las aplicaciones de su equipo.

Transferencia del conocimiento (5 min)

Equipo

Discuta:

Ofrecerá a los equipos la oportunidad de compartir sus prototipos entre ellos. Debido a que cada estudiante está trabajando en una versión separada de la aplicación del equipo, cada aplicación probablemente será ligeramente diferente. Alentará a los miembros del equipo a analizar sus diferentes soluciones para el mismo problema: ¿cuáles son los beneficios o los intercambios?

Experiencias de aprendizaje de profundización

Use estos Contenidos para ampliar el aprendizaje de los estudiantes. Se pueden usar como Contenidos extras fuera del aula.

Par de pantalla

Brindará a los estudiantes la oportunidad de practicar diseños rápidos y creación de prototipos emparejándolos para diseñar rápidamente y fusionar dos pantallas para una nueva aplicación.

Lluvia de ideas:

Como clase, pensarán en una lista de ideas de aplicaciones para fomentar una buena variedad.

Agrupación:

Colocará a los estudiantes en parejas, preferentemente con compañeros de un equipo diferente. Dará a cada par unos minutos para elegir una aplicación de la lista y acordará la pantalla que cada pareja diseñará, luego comenzarán un nuevo proyecto de App Lab. Después de un tiempo limitado dedicado al diseño de sus pantallas, los pares pueden intercambiar códigos compartidos e intentar importar las pantallas de los demás. El objetivo aquí no es crear una aplicación que funcione, sino practicar prototipos rápidos e iteración. A través de esta práctica, los estudiantes pueden comenzar a priorizar mejor lo que debe y lo que no debe entrar en una primera iteración.

Sugerencias de evaluación

Se sugiere los siguientes indicadores para evaluar formativamente los aprendizajes:

- Identifican mejoras a los problemas
- Planifican y priorizan las mejoras
- Gestionan roles de trabajo y comunican
- Documentan y explican el proceso de trabajo
- Reconocen los beneficios de la colaboración