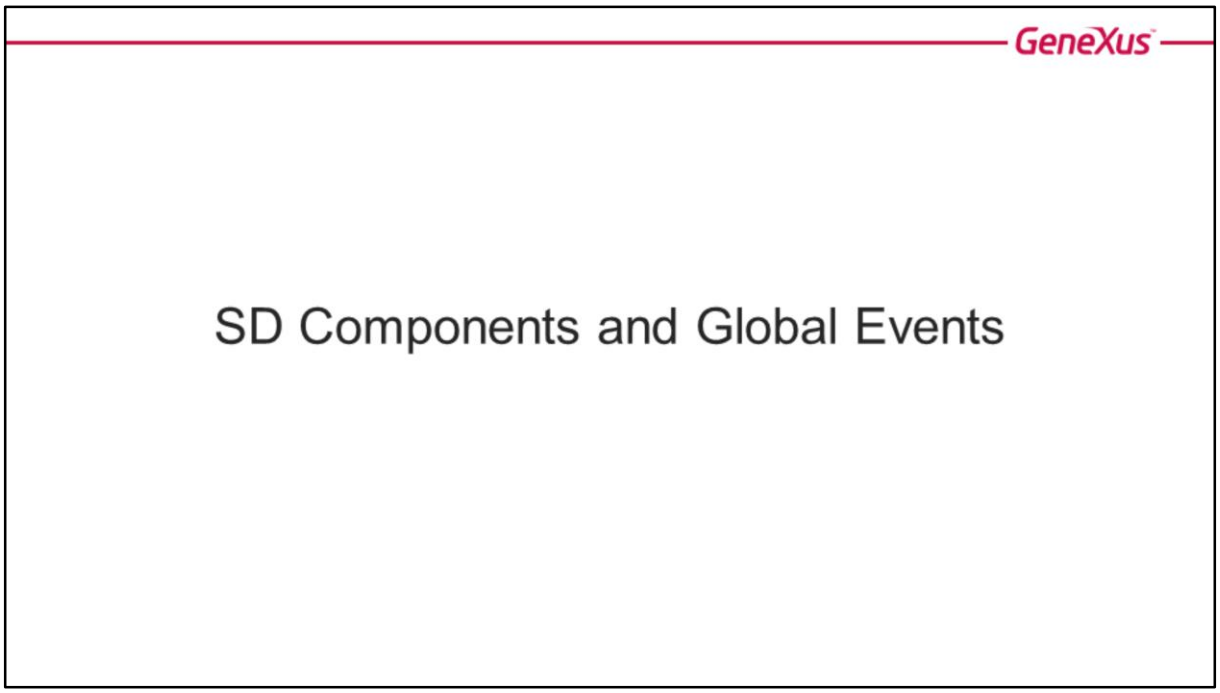
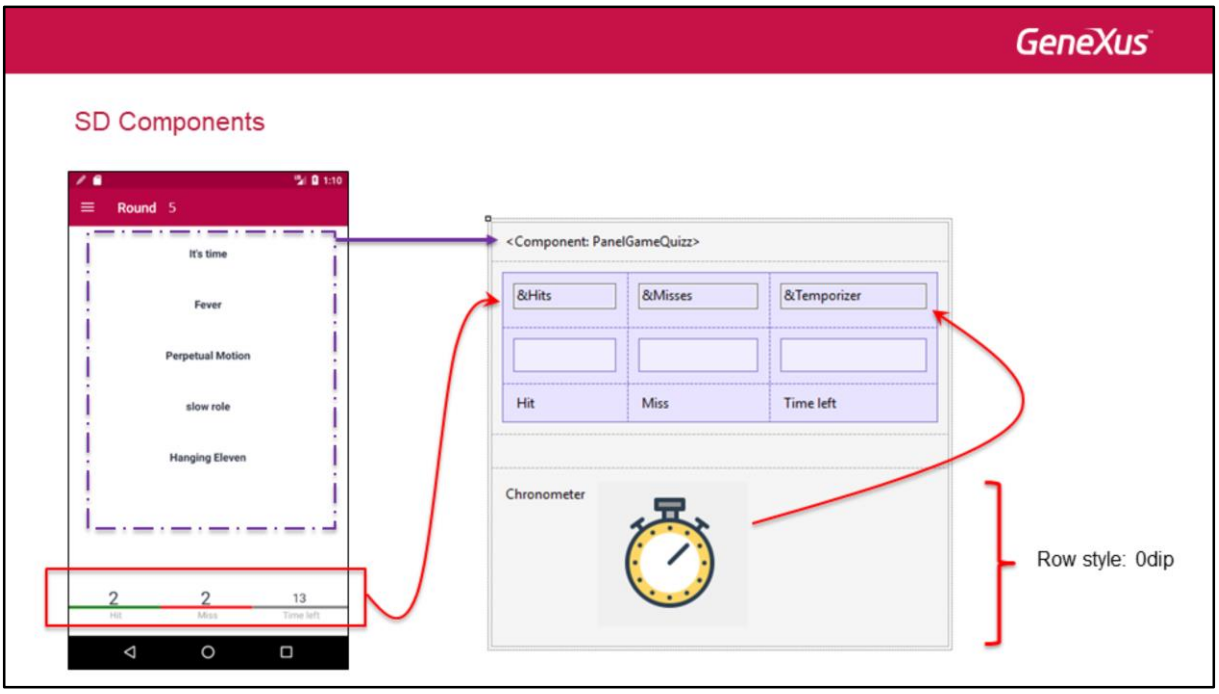
Smart Devices

GeneXus 15



Pasemos a un tema bien importante: SD Components and Global Events.



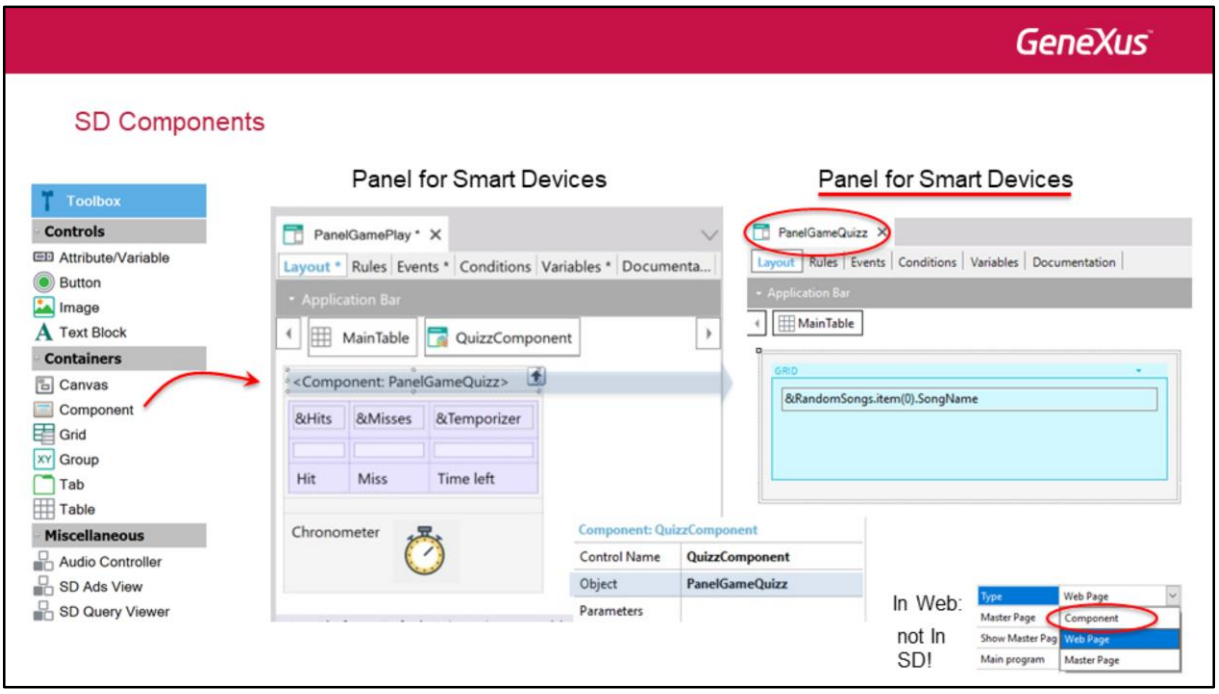
Aquí tenemos el panel PanelGamePlay que para cada ronda muestra las cinco canciones para que el usuario elija la que está sonando, y abajo se le muestra la cantidad de aciertos que tuvo hasta el momento en las rondas anteriores, de desaciertos, y el tiempo que le resta para adivinar la canción de la ronda actual.

Si vamos a observar el layout del panel, veremos que consta de un control nuevo, componente, que es como los componentes web, con algunas mínimas diferencias.

Básicamente, el concepto de Componente involucra algún tipo de “ventana” en un panel, a través de la cuál puede verse otro panel. En otras palabras, esto permite que el desarrollador pueda diseñar un panel que pueda ser ejecutado independientemente y pueda ser embebido en otro panel como componente de él.

Deberemos mantener sincronizados panel padre y panel hijo, puesto que cuando el usuario haga un tap en el componente (el panel que muestra las cinco canciones de la ronda actual), deberán actualizarse los contadores Hit y Miss, y además al cargarse una nueva ronda deberá resetearse el temporizador. De la misma manera, cuando se termine el tiempo de una ronda, deberá darse por finalizada y perdida.

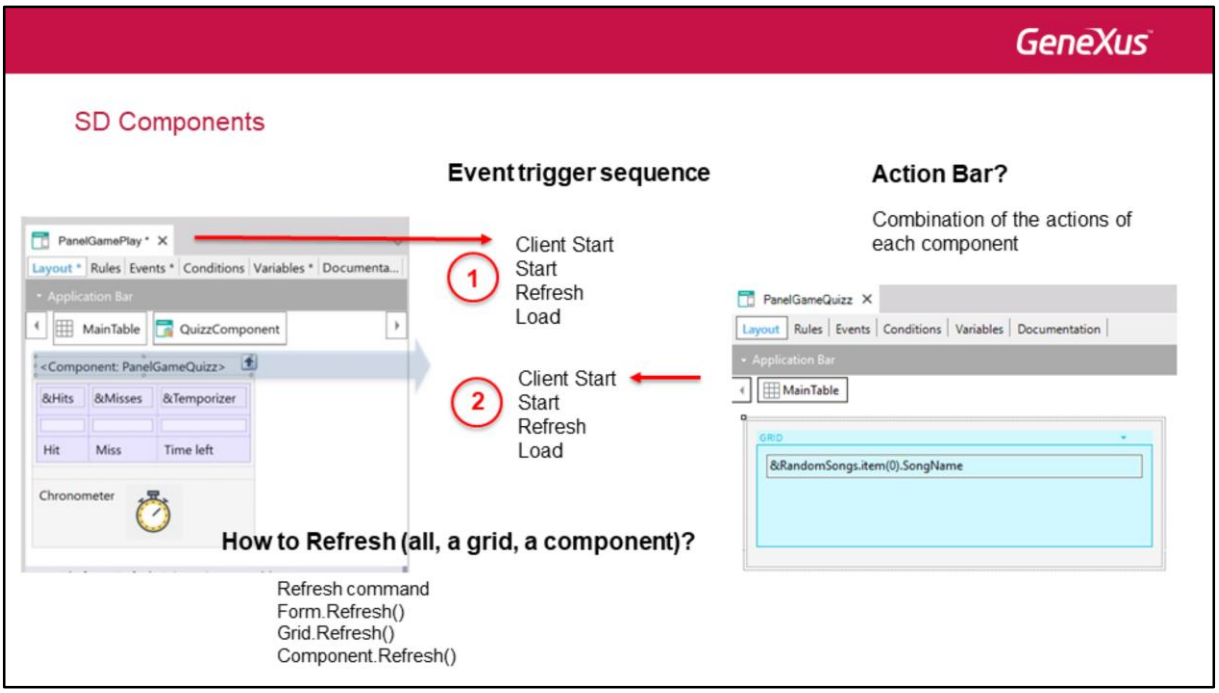
Para tener un timer necesitamos un nuevo tipo de control conocido como Chronometer, que deberá estar en la pantalla, aunque no quereamos visualizarlo. Como no puede utilizarse para él la propiedad Visible en 0, alcanzará con colocarlo en una fila con 0dips.



Observemos que se ha incorporado a la Toolbox el control Component. Una vez arrastrado al layout del Panel del que se trate (por supuesto puede ser también el layout de un Work With for Smart Devices), accediendo a sus propiedades se encontrará la de nombre **Object**, que es la que permite especificar qué objeto se cargará allí como componente. En nuestro caso cargaremos el Panel for Smart Devices PanelGameQuizz, que es el que carga las cinco canciones de la ronda actual. Nuestro PanelGameQuizz no tiene parámetros, pero si los tuviera, es en la propiedad **Parameters** del control Component donde éstos se especifican.

Estas propiedades, a diferencia de en Web, sólo pueden especificarse aquí, es decir, en el diseño. No pueden especificarse dinámicamente, en runtime.

Por otro lado, si bien en Web un panel componente debía especificarse explícitamente de ese tipo, en SD no. Cualquier Panel for Smart Devices, e incluso Work with for Smart Devices podrá ser ejecutado como componente.

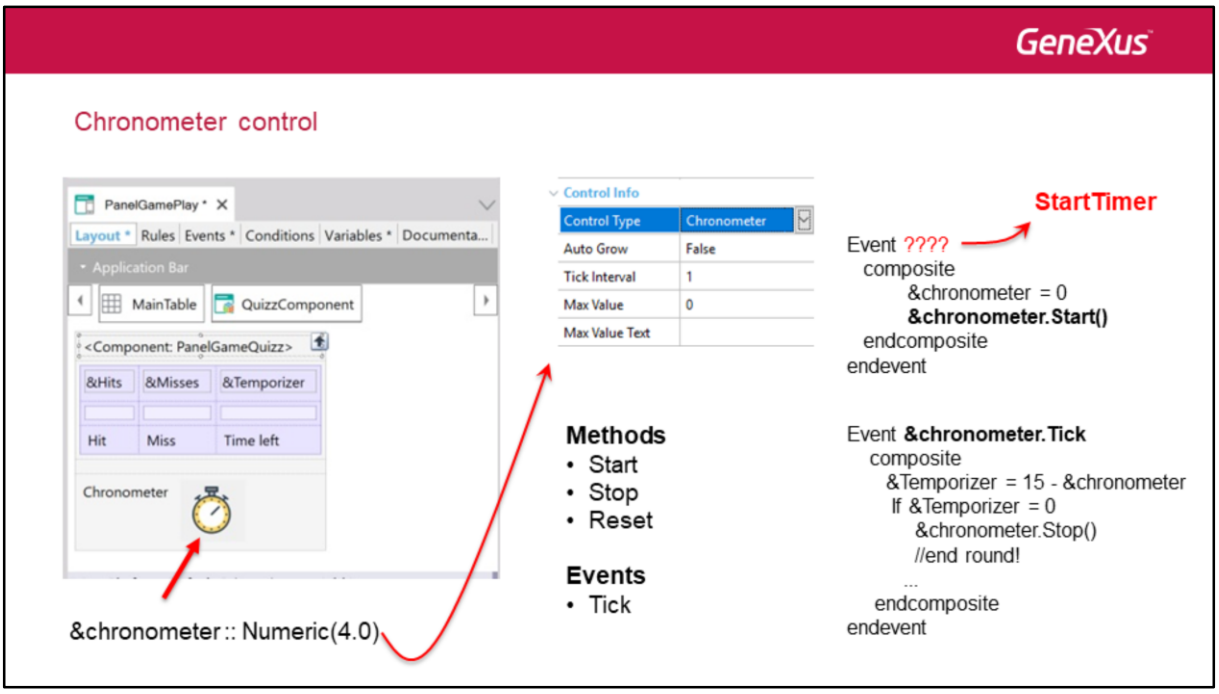


El orden de ejecución de los eventos cuando un panel tiene uno o más componentes es el mismo que el de la sección Detail de un Work with que tiene una o varias secciones.

Primero ejecuta los eventos asociados con el panel contenedor y luego los eventos de cada componente embebido, secuencialmente (uno por uno, en el orden en el que aparezcan en el layout).

¿Cómo refrescamos (todo, o un grid o un componente)? Tenemos los comandos y métodos que vemos aquí en pantalla. El comando Refresh. El Form punto Refresh, Grid punto Refresh o Component punto Refresh.

¿y qué pasa con la Action Bar? Si tenemos la Action Bar del padre y tenemos la action bar del contenido, del hijo. Bueno, se combinan las acciones de cada componente.



El control Chronometer nos da la posibilidad de ejecutar un evento luego de un cierto tiempo o simplemente mostrar un cronómetro en la pantalla. Basta con definir una variable o atributo numérico y cambiarle el tipo de control a Chronometer. Es válido tanto para Web como para SD.

Una vez hecho esto, se habilitan tres propiedades:

- Tick interval: indica la frecuencia en segundos en que se disparará el evento **Tick**.
- Max value: el máximo valor que el cronómetro puede tomar
- Max value text: el texto asociado a al variable cuando alcanza el valor máximo.

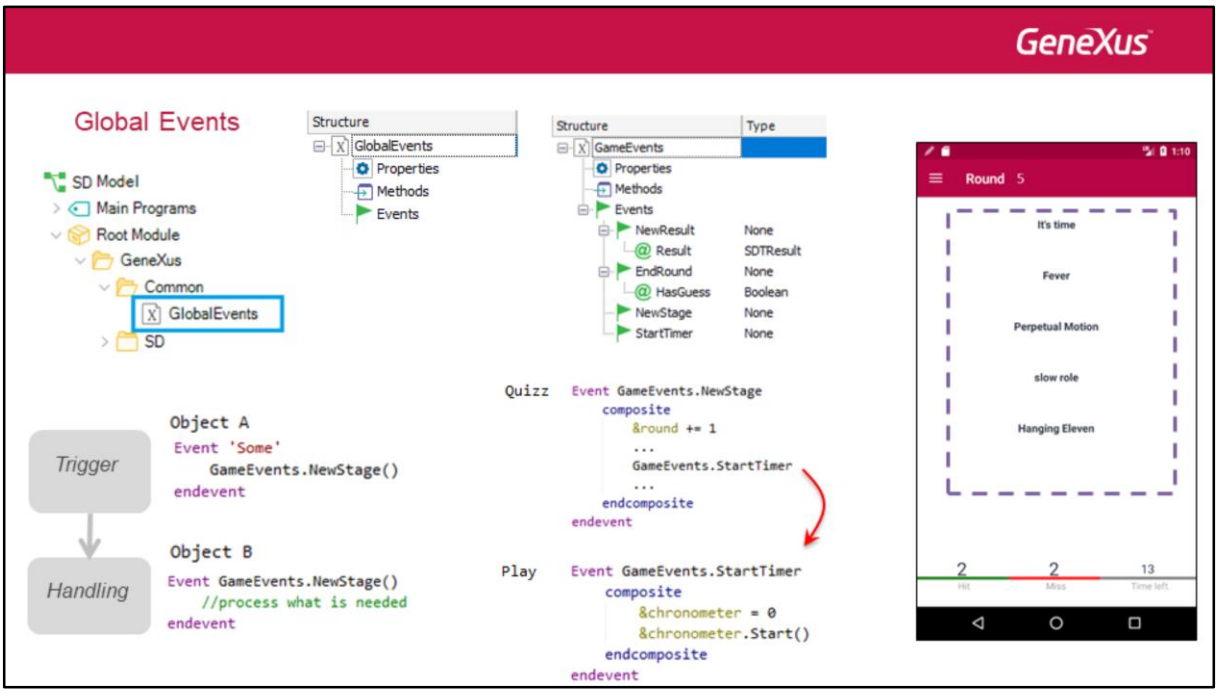
Además se habilitan tres métodos:

- Start: Inicia el cronómetro. Empezará por el valor (en segundos) contenido en el atributo/variable asociado al control.
- Stop: Detiene el cronómetro.
- Y Reset: Coloca en 0 el valor del cronómetro.

Y además y bien importante, se habilita el evento Tick que será ejecutado cada vez que el valor indicado en la propiedad Tick interval haya trascurrido.

En nuestro ejemplo, cuando inicia una nueva ronda del juego hay que iniciar el timer, es decir, el cronómetro debe arrancar en 0 y al llegar a 15 (o 20, lo que hayamos definido que es la duración de cada ronda del juego) detenerse porque se terminó el tiempo para esa partida.

A continuación veremos en qué evento colocar el Start del cronómetro. Será en un evento global que definiremos. Lo llamaremos StartTimer.



Como vimos cuando estudiamos lo nuevo en Web, en GeneXus 15 se ha incorporado un external object especial llamado **GlobalEvents**, que a diferencia de la mayoría de los external objects predefinidos, no viene en el módulo GeneXus, bajo References, sino dentro del folder GeneXus. La razón: será modificado por el desarrollador. Para eso fue hecho.
No obstante, podemos salvarlo con otro nombre, y tener un external object de nombre específico, por ejemplo GameEvents, para nuestra aplicación.

Aquí el desarrollador podrá definir eventos que lograrán comunicar a paneles, por ejemplo para sincronizar acciones entre sí.
En nuestro ejemplo hemos definido cuatro eventos globales para poder sincronizar al panel PanelGamePlay con su hijo PanelGameQuizz.

Alguno o algunos de los paneles dispararán el evento, que será escuchado por el o los paneles que lo tengan definido como evento en su sección Events. Estos eventos pueden o no tener parámetros.

De hecho, un mismo panel puede disparar un evento y manejarlo (¡Object A y Object B pueden ser el mismo!). Así, desde un evento de un panel puede ejecutarse otro evento del mismo panel, invocándolo.

Este es un uso simple de la potencia de los GlobalEvents, pero también pueden ser utilizados en paneles independientes (no necesariamente para sincronizar componentes y el panel padre).

A la derecha vemos un ejemplo: el panel componente PanelGameQuizz en un evento global NewStage que será disparado desde el contenedor PanelGamePlay (aquí no lo mostramos) cuando hay que iniciar una nueva ronda de canciones, incrementa el número de ronda en 1 y dispara la inicialización del timer, evento global que es manejado por el PanelGamePlay, que es el que tiene el cronómetro y muestra el tiempo restante (a partir del evento Tick que vimos antes).

Con esto vimos unos ejemplos de uso de los Global Events en combinación con los componentes. Para ver más sobre todo esto, acceda a nuestro wiki.

More about external objects