

Problemas y workarounds para el desarrollo de aplicaciones Ginga

Fecha: 19-08-2015

Revisión: 1

Contenido

1	Objetivo	2
2	Receptores	3
3	Problemas comunes	4
3.1	Directorio relativo “../”	4
3.2	Codificación de caracteres	4
3.3	Texto con múltiples líneas.....	5
3.4	Medición de alto y ancho de texto con múltiples líneas.....	5
3.5	Composición de canvases	6
3.6	Chroma key con funciones de dibujo.....	6
3.7	Chroma key con imágenes	7
3.8	Colores con transparencia parcial.....	7
3.9	Combinación de colores con transparencia parcial	8
3.10	Limpiar región del canvas.....	8
3.11	Composición de canvases con parámetros adicionales opcionales.....	9
3.12	Variables settings	9
3.13	Recorte de región de dibujo en canvas.....	10
3.14	Instanciación de canvas con un archivo inexistente	10
3.15	Fuentes de texto	11
3.16	Conexión TCP fallida.....	12
3.17	Reproducción de audio	12
3.18	Reproducción de video.....	12
3.19	Resolución de la capa gráfica Ginga	13
4	Problemas particulares	14
4.1	La aplicación no ejecuta	14
4.2	Configuración de fuente de texto y medición de texto en canvas	14
4.3	Obtención del estilo de la fuente configurada.....	15
4.4	Doble flush de canvas.....	15
4.5	Variables settings mal definidos	15
4.6	Escalado de la capa de video de la señal.....	16
4.7	Teclas numéricas	16
4.8	Tecla “PAUSA”	16
4.9	Escala y posterior recorte de canvas.....	16
4.10	Ejecución automática luego de cerrar aplicación	17
4.11	Eventos de teclas.....	17
4.12	Composición de canvas con región de dibujo recortada	17
4.13	Eventos de tecla espurios.....	17

1 Objetivo

En este documento se describen anomalías y diferencias encontradas en la implementación del middleware Ginga de varios receptores. Se proponen soluciones a los problemas que surgen a partir de estas diferencias, en caso que sea posible.

Para referencias acerca de especificaciones del middleware, referirse a: ABNT NBR 15606, en particular, partes 1 y 2.

2 Receptores

A continuación se presentan todos los receptores referidos en este documento:

ID	Receptor
01	CORADIR 1000D
02	CORADIR 1005D
03	CORADIR 1800D
04AR	CORADIR 3000D (fw 1.0, sintonización Argentina)
04BR	CORADIR 3000D (fw 1.0, sintonización Brasileira)
05AR	CORADIR 3000D (fw 1.09, sintonización Argentina)
05BR	CORADIR 3000D (fw 1.09, sintonización Brasileira)
06	UTE 740 (fw v2.2.6)
07	UTE 740 (fw v2.3.6)
08	UTE 742A (fw v3.1.18)
09	NEWTRONIC DV-5306-SATVD-T (fw FDTCGTLA v2.2)
10	NEWTRONIC DV-5306-SATVD-T (fw FDTCGTLA v2.4)
11	NEWTRONIC DV-5306-SATVD-T (fw zapper 018.032.002)
12	KAON FR9100 (fw v01.10.00.01)
13	KAON FR9100 (fw v01.10.00.02)
14	VM Ginga.ar 1.2
15	VM Ginga.ar 1.3
16	VM Ginga.ar 2.0
17	VM Ginga.ar 2.1
18	VM Ginga.ar 2.2

3 Problemas comunes

En esta sección se describirán algunos problemas que comparten varios receptores. Para cada problema se propone una forma de evitarlo, en caso que sea posible, y se listan los receptores afectados.

3.1 Directorio relativo “../”

Problema

Algunos receptores no soportan la referencia a archivos que estén fuera del directorio en el que se encuentra el archivo .lua ejecutado por el media NCL.

Receptores afectados

01, 03, 04AR, 04BR, 05AR, 05BR, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18

Solución

Evitar la referencia a archivos fuera del directorio en la que se encuentra el archivo .lua. Organizar los archivos que necesite utilizar un media LUA en carpetas incluidas dentro de la misma ruta del archivo .lua de acceso.

3.2 Codificación de caracteres

Problema

Se utiliza codificación de caracteres UTF8, distinta a la especificada por la norma. La codificación de caracteres requerida en Ginga parte de la ISO/IEC 8859-15:1999, con algunas modificaciones (ver ABNT NBR 15606-1:2013, sección 11.4). La codificación de caracteres distinta a la esperada suele provocar conflictos al utilizarlos con funciones que operan con cadenas de caracteres, por ejemplo: *canvas:drawText*, *canvas:measureText* de Ginga.

Receptores afectados

02, 03, 04AR, 05AR, 06, 07, 08, 11, 12, 13, 14, 15, 16, 17, 18

Solución

Una solución es identificar el receptor y realizar la conversión a la codificación que soporta.

3.3 Texto con múltiples líneas

Problema

La función `canvas:drawText(x, y: number, text:string)` en algunos dispositivos soporta textos con caracteres de nueva línea (`'\n'`) y en otros no.

Receptores afectados

No soportan múltiples líneas: 14, 15

Solución

Para contar con la posibilidad de dibujar en pantalla un texto en múltiples líneas, una alternativa es implementar un método que busque los caracteres de nueva línea en la cadena, y dibuje cada línea por separado utilizando el método `canvas:drawText` múltiples veces (una vez por cada carácter `'\n'` encontrado, además del último caso en el que no se encuentran más caracteres de ese tipo).

3.4 Medición de alto y ancho de texto con múltiples líneas

Problema

Similar al caso anterior (del texto de múltiples líneas), la función `canvas:measureText(text:string)` en algunos dispositivos considera los caracteres de nueva línea (`'\n'`) y en otros no.

Receptores afectados

14, 15

Solución

Análogo al caso del `canvas:drawText`, para contar con la posibilidad de medir en píxels textos que luego se dibujen en pantalla en múltiples líneas, una alternativa posible es implementar un método que busque los caracteres nueva línea en la cadena, y utilice `canvas:measureText` múltiples veces (una vez por cada carácter `'\n'` encontrado, además del último caso en el que no se encuentran más caracteres de ese tipo). El resultado que devuelve podría ser el máximo ancho de las líneas medidas y el alto acumulado entre todas.

3.5 Composición de canvases

Problema

El método *canvas:compose(x, y: number, src: canvas)* sólo funciona correctamente si el canvas fuente (*src*) es una imagen. Por ejemplo, si se instancia un canvas *nuevoCanvas*, y se dibuja luego un texto en el mismo, llamando al método *nuevoCanvas:drawText(x, y, text)*, y luego se lo compone con el principal, el texto no se visualiza, o se deforma. En un caso similar, el método *canvas:drawRect* sobre un canvas instanciado presenta problemas.

Receptores afectados

03, 04AR, 05AR, 06, 07, 08, 12, 13

Solución

Restringir el uso de la función *canvas:compose* a casos de composición del canvas principal con una imagen.

3.6 Chroma key con funciones de dibujo

Problema

Los colores magenta ($r = 255, g = 0, b = 255$) y/o negro ($r = 0, g = 0, b = 0$) se convierten a transparente (se puede ver el video del canal donde se dibuja con este color) al utilizarlos para graficar en la pantalla con las funciones de dibujo (por ejemplo *drawRect* o *drawText*).

Receptores afectados

drawRect y negro: 01, 03, 09, 10

drawRect y magenta: 06, 07, 08, 11

drawText y negro: 03, 11

drawText y magenta: 06, 07, 11

Solución

Evitar el uso de magenta y/o negro puros, reemplazando, por ejemplo, el negro por $r = 1, g = 0, b = 0$.

3.7 Chroma key con imágenes

Problema

Los colores magenta ($r = 255, g = 0, b = 255$) y/o negro ($r = 0, g = 0, b = 0$) de una imagen se convierten a transparente (se puede ver el video del canal donde se dibuja con este color) al utilizarlos para componer con el canvas principal, dependiendo del formato de archivo de la imagen.

Receptores afectados

PNG y negro: 01, 02, 03, 06, 07, 08, 09, 10, 11

PNG y magenta: 06, 07, 11

JPG y negro: 01, 06, 07, 09, 10, 11

JPG y magenta: 06, 07, 11

Solución

Evitar el uso de magenta y/o negro puros en imágenes, reemplazando, por ejemplo, el negro por $r = 1, g = 0, b = 0$.

3.8 Colores con transparencia parcial

Problema

Se convierten los colores con transparencia parcial (canal alpha con valor entre 1 y 254) a colores totalmente opacos, y con un cierto tinte de magenta, cuando se los dibuja sobre la capa del video de la señal del canal.

Receptores afectados

06, 07, 11, 14

Solución

Evitar dibujar colores con transparencias parciales sobre la capa del video de la señal del canal.

3.9 Combinación de colores con transparencia parcial

Problema

La mezcla de colores semitransparentes no funciona. Es decir, en la región pintada solamente se verá el último color utilizado.

Receptores afectados

14, 15

Solución

No superponer colores semitransparentes, para lograr que la pantalla se vea igual en todos los receptores.

3.10 Limpiar región del canvas

Problema

Al utilizar `canvas:clear(x, y, width, height: number)`, se ignoran los parámetros provistos. Es decir que el resultado es igual que llamar al método `canvas:clear()`.

Receptores afectados

14, 15

Solución

El conjunto de dispositivos que tienen este problema comparten además una incapacidad de fusionar colores semitransparentes. Por lo tanto, para este conjunto de dispositivos se puede utilizar `canvas:drawRect('fill', x, y, width, height)` en lugar del método `canvas:clear(x, y, width, height)`. Por ejemplo, si se quiere borrar un sector de la pantalla, se puede configurar el color transparente con `canvas:attrColor(0,0,0,0)` y luego `canvas:drawRect('fill', x, y, width, height)`.

3.11 Composición de canvases con parámetros adicionales opcionales

Problema

Algunos receptores no implementan (ignoran) los parámetros opcionales del método `canvas:compose(x, y: number, src: canvas; [src_x, src_y, src_width, src_height: number])`.

Receptores afectados

14, 15

Solución

Para lograr el comportamiento del método `canvas:compose` con parámetros opcionales como se encuentra descrito en la norma, se puede utilizar la función `canvas:attrClip` de la siguiente manera:

```
local clip_x, clip_y, clip_w, clip_h = canvas:attrClip()  
canvas:attrClip(x, y, src_width, src_height)  
canvas:compose(x - src_x, y - src_y, src)  
canvas:attrClip(clip_x, clip_y, clip_w, clip_h)
```

3.12 Variables settings

Problema

El objeto settings de Ginga no es accesible desde un script LUA que ejecuta el media NCL.

Receptores afectados

01, 02, 03, 04AR, 06, 07, 08, 11, 12, 13, 14, 15

Solución

Acceder al objeto settings por medio de NCL.

El listado de variables accesibles, ya definidas, se encuentra especificado en la norma ABNT NBR 15606-2:2011, Tabla 12.

Se debe declarar el media especial de tipo “application/x-ginga-settings”, específico para manejar variables globales del documento NCL. Por ejemplo, para acceder a la variable `system.memory`, es necesario definir:

```
<media id="settings" type="application/x-ginga-settings">  
  <property name="system.memory"/>  
</media>
```

Para copiar valores de variables de un media a otro se debe utilizar conectores NCL.

3.13 Recorte de región de dibujo en canvas

Problema

Se interpretan de forma diferente a la norma los parámetros del método `canvas:attrClip(x, y, width, height: number)`.

El método toma dos valores x , y que conforman una coordenada en pantalla, y $width$, $height$ como el ancho y alto del recorte de la región de dibujo del `canvas`.

Algunos receptores implementan el método como `canvas:attrClip(x1, y1, x2, y2: number)`, donde los parámetros x_1 , y_1 son la coordenada superior izquierda y x_2 , y_2 la coordenada inferior derecha de la región de recorte de dibujo del `canvas`.

Receptores afectados

03, 04AR, 05AR, 06, 07, 08, 09, 10, 11, 14, 15

Solución

Identificar los receptores y, cuando corresponda, aplicar la transformación

```
x = x
y = y
width = x + width - 1
height = y + height - 1
```

para asignar los valores de clipping, y

```
x = x
y = y
width = width - x + 1
height = height - y + 1
```

para devolver los valores de clipping asignados al `canvas`.

3.14 Instanciación de canvas con un archivo inexistente

Problema

El método `canvas:new(image_path: string)` en algunos receptores cierra la aplicación de forma inesperada si el archivo no existe, y en otros arroja una excepción que puede manipularse utilizando el llamado a función en modo protegido (*pcall*).

Receptores afectados

Salida inesperada: 14, 15

Solución

Evitar intentar instanciar un canvas con una ruta de un archivo inexistente.

3.15 Fuentes de texto

Problema

No se implementan las fuentes de texto obligatorias por norma: 'Tiresias' y 'Verdana', con estilo normal y 'bold', 'italic' o 'bold-italic'.

Receptores afectados

Las celdas pintadas de gris corresponden al caso tal que aquel dispositivo que no tiene implementada la fuente de texto:

	01	02	03	04	05	06	07	08	09	10	11	12	13
	AR AR												
Tiresias' normal													
Tiresias' 'bold'													
Tiresias' 'italic'													
Tiresias' 'bold-italic'													
Verdana' normal													
Verdana' 'bold'													
Verdana' 'italic'													
Verdana' 'bold-italic'													

Solución

Evitar utilizar fuente 'Verdana' o los estilos 'bold', 'italic' o 'bold-italic' de 'Tiresias'. En su lugar, utilizar 'Tiresias' con estilo normal.

3.16 Conexión TCP fallida

Problema

Luego de un intento de conexión fallido por canal de retorno, el evento resultante no tiene valor en el campo *error*, y sí tiene valor en el campo *connection*. Esto no permite diferenciar un intento de conexión fallido de uno válido.

Comportamiento esperado

Al realizar un intento de conexión por canal de retorno (TCP), se recibe un evento NCL de la forma:

```
{ class='tcp', type='connect', host=addr, port=number, connection=identifier, error=<err_msg>}
```

Los campos *error* y *connection* son mutuamente exclusivos. Es decir, que si la conexión fue válida, se debería recibir un valor en el campo *connection* y ninguno en *error*, y si la conexión fue fallida, se debería recibir un valor en *error* y ninguno en *connection*.

Receptores afectados

03, 04AR, 04BR, 06, 07, 08, 11, 12, 13, 14, 15

Solución

En estos casos el identificador de conexión (recibido en el evento) es un objeto de clase 'LuaSocket'.

Un objeto LuaSocket tiene un método *getpeername()* que devuelve el IP y puerto de la conexión, en caso de haber sido válida. Si no, devuelve *nil*.

Sabiendo esto, se puede utilizar el método *getpeername()* para diferenciar una conexión válida de una fallida.

3.17 Reproducción de audio

Problema

La reproducción de audio no funciona en las aplicaciones Ginga.

Receptores afectados

14, 15, 18

3.18 Reproducción de video

Problema

La reproducción de video no funciona en las aplicaciones Ginga.

Receptores afectados

18

3.19 Resolución de la capa gráfica Ginga

Problema

La resolución en la que ejecuta el plano gráfico de Ginga difiere entre los dispositivos, y depende además de la señalización de la capa de transporte configurada para el canal. Algunos casos encontrados (para el mismo canal y con la misma configuración de transporte) son: 720x480, 720x576 y 1280x720. Algunos receptores siempre ejecutan el plano gráfico de Ginga en resolución 720x576.

Receptores afectados

Siempre en 720x576: 01, 03, 04AR, 04BR, 05AR, 04BR, 06, 07, 08, 11, 12, 13

4 Problemas particulares

En esta sección se describirán algunos problemas particulares de algunos receptores. Para cada problema se propone una forma de evitarlo, en caso que sea posible, y se listan los receptores afectados.

4.1 La aplicación no ejecuta

Problema

La aplicación no ejecuta con la plataforma actual de transmisión de señal de los canales.

Receptores afectados

01, 09, 10

4.2 Configuración de fuente de texto y medición de texto en canvas

Problema

La aplicación cierra inesperadamente si se alterna reiteradamente el uso del método *canvas:attrFont(face: string, size: number, style: string)* -para realizar cambios en la configuración de la fuente (por ejemplo, su tamaño)- con la función *canvas:measureText(text: string)* -para medir el tamaño de un texto en pantalla.

Receptores afectados

04AR, 05AR, 06, 07, 12, 13

Solución

Dos alternativas posibles serían:

- a) Utilizar la misma fuente de texto para toda la aplicación. Esto implica que el método *canvas:attrFont* se llama una única vez, y luego se pueden hacer múltiples mediciones de texto en pantalla con *canvas:measureText*.
- b) Utilizar múltiples fuentes de texto y evitar la utilización de la función *canvas:measureText*.

4.3 Obtención del estilo de la fuente configurada

Problema

Al consultar la fuente de texto configurada con el método *canvas:attrFont*, el segundo valor devuelto (estilo de la fuente) comienza con mayúscula, y es distinto al valor que se puede configurar.

Receptores afectados

16

Solución

Convertir a minúscula el segundo valor devuelto por el método *canvas:attrFont*.

4.4 Doble flush de canvas

Problema

Al dibujar y llamar al método *canvas:flush()*, los cambios no se reflejan en pantalla hasta el próximo *canvas:flush()*.

Receptores afectados

06

Solución

Llamar a *canvas:flush()* dos veces siempre que se requiera reflejar los cambios en pantalla.

4.5 Variables settings mal definidos

Problema

El nombre de algunas variables del objeto settings se encuentran mal definidos. Por ejemplo, *system.GingaNCL.version* está cambiado por *system.gingaNCLVersion*, *system.classType[1]* está cambiado por *system.classType*, y *system.ncl.version* por *system.nclVersion*.

Receptores afectados

05AR, 05BR

4.6 Escalado de la capa de video de la señal

Problema

El escalado de la capa de video de la señal del canal no funciona correctamente. Al hacerlo, resulta en una escala distinta a la esperada, y no permite volver al tamaño original.

Receptores afectados

01, 04BR, 05BR

Solución

No utilizar escalado de la capa de video de la señal del canal en estos equipos.

4.7 Teclas numéricas

Problema

Las teclas numéricas no son capturadas por la aplicación.

Receptores afectados

04AR

4.8 Tecla "PAUSA"

Problema

El control remoto del dispositivo no tiene tecla "VOLVER", y en su lugar la tecla "PAUSA" funciona como tal.

Receptores afectados

05AR, 05BR

4.9 Escala y posterior recorte de canvas

Problema

Al realizar la escala de un *canvas* con el método *canvas:attrScale* y luego el recorte del mismo, con la función *canvas:attrCrop*, el resultante es un *canvas* recortado, pero de dimensiones superiores a la escala configurada.

Receptores afectados

05AR, 05BR, 11

4.10 Ejecución automática luego de cerrar aplicación

Problema

Al presionar la tecla EXIT se cierra la aplicación, pero luego se vuelve a ejecutar.

Receptores afectados

12

4.11 Eventos de teclas

Problema

No se recibe el evento 'release', que indica el momento en que se suelta la tecla del control remoto.

Receptores afectados

12, 13

4.12 Composición de canvas con región de dibujo recortada

Problema

Al realizar una composición de un *canvas* en otro, la configuración de la región de dibujo (*canvas:attrClip*) del *canvas* destino de la composición se reconfigura al tamaño completo.

Receptores afectados

16

4.13 Eventos de tecla espurios

Problema

El dispositivo ocasionalmente envía eventos de teclas espurios (distintos a la tecla que se está presionando). Por ejemplo: si presionamos varias veces el botón de FLECHA ABAJO, podría enviarse un evento de tecla AMARILLA.

Receptores afectados

03