Práctica 3
Uso de la imagen en la videoconsola 3DS

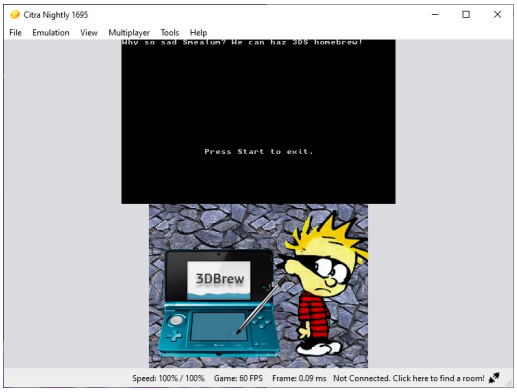
Padró Ferragut, Cristina

Mapas de bits: acceso directo a la memoria del vídeo

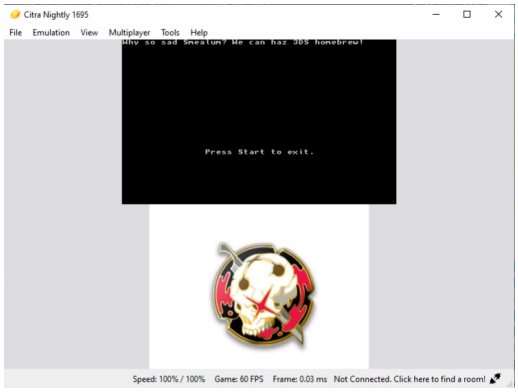
Ejercicio 1

Compruebe que puede sobreescribir el fichero PNG mencionado, dentro del subdirectorio `gfx`, con cualquier fichero gráfico de su elección y que, al reconstruir el proyecto, esa nueva imagen aparece en la pantalla inferior.

Si cambiamos la imagen dada `brew.png` y ponemos otra imagen de 240x320 y la llamamos `brew.png`, al compilar, cambiará la imagen correctamente.



(a) Ejemplo de `graphics/bitmap/24bit-color`



(b) Modificación de `graphics/bitmap/24bit-color`

Ejemplo de acceso directo a memoria

Ejercicio 2

Compruebe en el código entregado que los `buffer` se crean con tipo `GSP_RGBA8_OES` (esto indicará que se utilizan 32 bits para cada píxel, 8 por cada componente de RGB, más el canal alfa de transparencia) y que el `buffer` está rotado respecto a las coordenadas originales. Anote cómo se ve este último aspecto en el código que inicializa los `buffers` de la memoria.

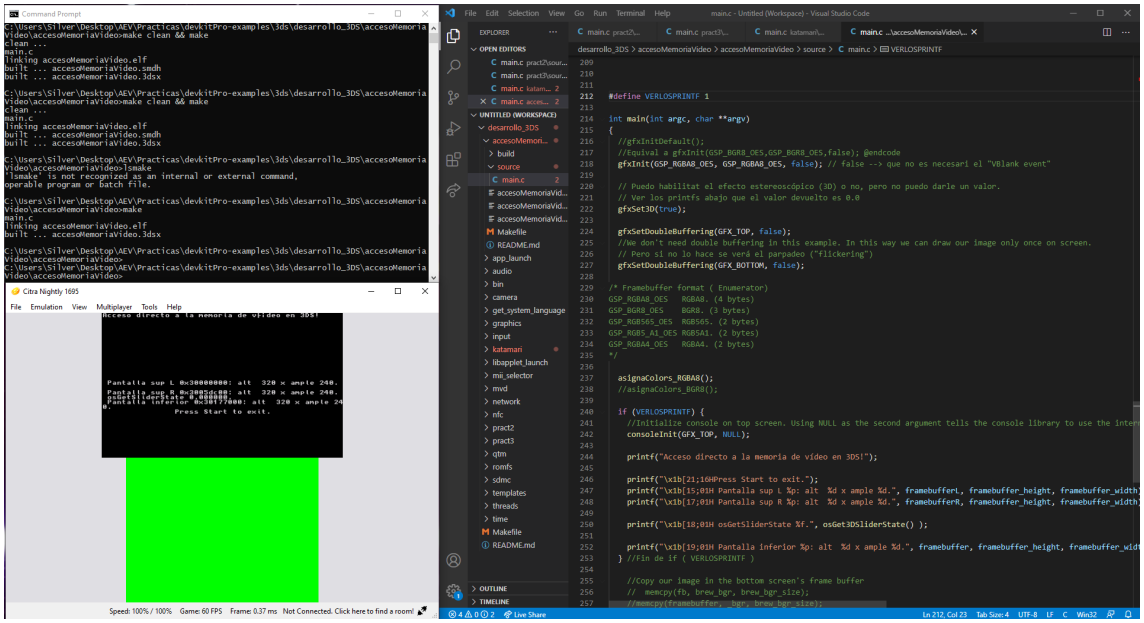


Figure 2: Codigo y ejecución de `accesoMemoriaVideo`

Gráficas en 2D con Citro2D

Gestión avanzada del texto

Ejercicio 3

Sobre el ejemplo de 3DS `graphics/printing/system-font`, comprobará que si es posible utilizar caracteres como la ñ del español. Veamos si también es posible hacer uso de la ç del valenciano, las tildes acentuadas o la diéresis de estos idiomas.

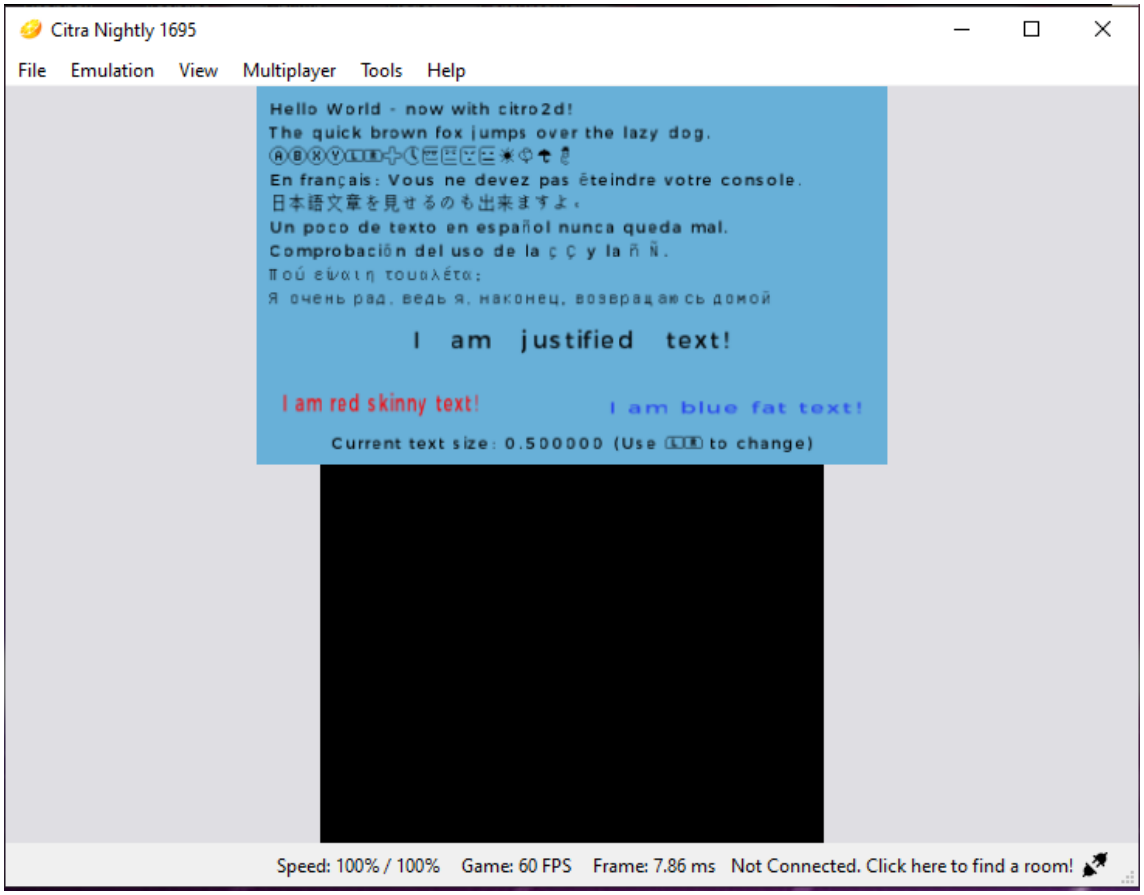


Figure 3: Añadidos de pruebas de Ç y Ñ en `graphics/printing/system-font`

Primitivas de dibujo 2D

Ejercicio 4

Sobre el ejemplo de 3DS `graphics/printing/2d_shapes`, anote cómo han dibujado los tres círculos para que cada uno haya salido con un color o degradado de color de relleno. Un apunte al respecto del dibujo de primitvas, lo dice el código de este ejemplo y, también lo dice la documentación, es que el orden de dibujado de las primitivas gráficas puede ser importante.

```
// Create colors
u32 clrWhite = C2D_Color32(0xFF, 0xFF, 0xFF, 0xFF);
u32 clrGreen = C2D_Color32(0x00, 0xFF, 0x00, 0xFF);
u32 clrRed   = C2D_Color32(0xFF, 0x00, 0x00, 0xFF);
u32 clrBlue  = C2D_Color32(0x00, 0x00, 0xFF, 0xFF);

u32 clrCircle1 = C2D_Color32(0xFF, 0x00, 0xFF, 0xFF);
u32 clrCircle2 = C2D_Color32(0xFF, 0xFF, 0x00, 0xFF);
u32 clrCircle3 = C2D_Color32(0x00, 0xFF, 0xFF, 0xFF);

u32 clrSolidCircle = C2D_Color32(0x68, 0xB0, 0xD8, 0xFF);

u32 clrTri1 = C2D_Color32(0xFF, 0x15, 0x00, 0xFF);
u32 clrTri2 = C2D_Color32(0x27, 0x69, 0xE5, 0xFF);

u32 clrRec1 = C2D_Color32(0x9A, 0x6C, 0xB9, 0xFF);
u32 clrRec2 = C2D_Color32(0xFF, 0xFF, 0x2C, 0xFF);
u32 clrRec3 = C2D_Color32(0xD8, 0xF6, 0x0F, 0xFF);
u32 clrRec4 = C2D_Color32(0x40, 0xEA, 0x87, 0xFF);

u32 clrClear = C2D_Color32(0xFF, 0xD8, 0xB0, 0x68);
```

(a) Codigo de `graphics/printing/2d_shapes`

```
// Render the scene
C3D_FrameBegin(C3D_FRAME_SYNCDRAW);
C2D_TargetClear(top, clrClear);
C2D_SceneBegin(top);

C2D_DrawTriangle(50 / 2, SCREEN_HEIGHT - 50, clrWhite,
    0,  SCREEN_HEIGHT, clrTri1,
    50, SCREEN_HEIGHT, clrTri2, 0);
C2D_DrawRectangle(SCREEN_WIDTH - 50, 0, 0, 50, 50, clrRec1, clrRec2, clrRec3, clrRec4);

// Circles require a state change (an expensive operation) within citro2d's internals, so draw them last.
// Although it is possible to draw them in the middle of drawing non-circular objects
// (sprites, images, triangles, rectangles, etc.) this is not recommended. They should either
// be drawn before all non-circular objects, or afterwards.
C2D_DrawEllipse(0, 0, 0, SCREEN_WIDTH, SCREEN_HEIGHT, clrCircle1, clrCircle2, clrCircle3, clrWhite);
C2D_DrawCircle(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, 0, 50, clrCircle3, clrWhite, clrCircle1, clrCircle2);
C2D_DrawCircle(25, 25, 0, 25,
    clrRed, clrBlue, clrGreen, clrWhite);
C2D_DrawCircleSolid(SCREEN_WIDTH - 25, SCREEN_HEIGHT - 25, 0, 25, clrSolidCircle);
C3D_FrameEnd(0);
```

(b) Codigo de `graphics/printing/2d_shapes`

Sprites y animaciones con Citro 2D

Ejercicio 5

Sobreescriba los dos primeros PNG de la lista de `sprites.t3s`, en el ejemplo de `gpusprites`, para comprobar que, al reconstruir el proyecto, estas nuevas imágenes aparecerán en el lugar de las anteriores. Baje el número de `sprites` en pantalla para observarlo con facilidad y haga una captura de pantalla.

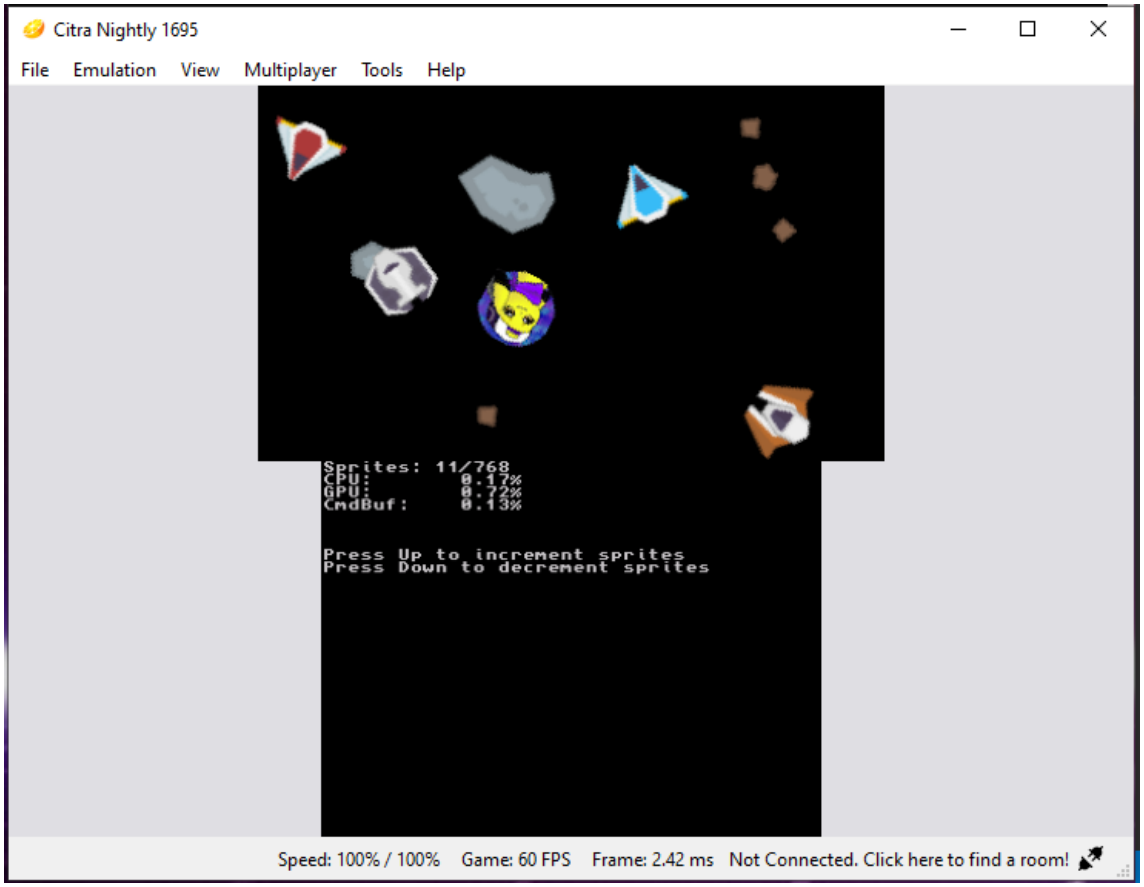
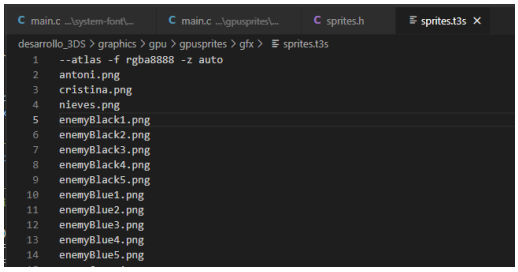
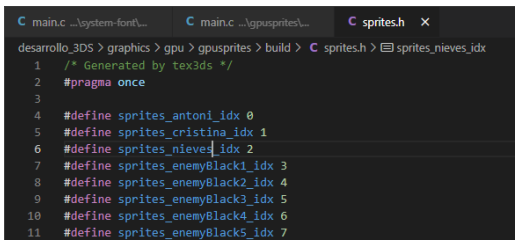


Figure 5: Sprite cambiado en `gpusprites`



(a) Codigo cambiado para `gpusprites` en `sprites.t3x`



(b) Codigo cambiado para `gpusprites` en `sprites.h`

Dibujo en 3D con Citro 3D

Ejercicio 6

Describa el contenido del proyecto del ejemplo para 3DS `graphics/gpu/texture_cube`, indicando qué archivos existen de partida y cuáles se generan con la construcción del ejecutable.

Los archivos originales son:

- **Carpeta source**: `main.c`, `vshader.v.pica`.
- **Carpeta gfx**: `kitten.png`, `kitten.t3s`.
- `Makefile`

Se crean los archivos:

- **Carpeta build**: `kitten.d`, `kitten.h`, `kitten.t3x`, `kitten.t3x.o`, `kitten.t3x.h`, `main.d`, `main.o`, `textured_cube.lst`, `textured_cube.map`, `vshader.shbin`,`vshader.shbin.d` `vshader.shbin.o`, `vshader.shbin.h`.
- `textured_cube.3dsx`
- `textured_cube.elf`
- `textured_cube.smdh`

Ejercicio 7

Describa el contenido del proyecto del ejemplo para 3DS `graphics/gpu/cubemap`, indicando qué archivos existen de partida y cuáles se generan con la construcción del ejecutable.

Los archivos originales son:

- **Carpeta source**: `main.c`, `skybox.v.pica`.
- **Carpeta gfx**: `skybox.png`, `skybox.t3s`.
- `Makefile`

Se crean:

- **Carpeta build**: `skybox.d`, `skybox.h`, `skybox.t3x.o`, `skybox.t3x.h`, `main.d`, `main.o`, `cubemap.lst`, `cubemap.map`, `skybox.shbin`,`skybox.shbin.d`, `skybox.shbin.o`, `skybox_shbin.h`.
- `cubemap.3dsx`
- `cubemap.elf`
- `cubemap.smdh`