CS 134

Text Drawing &
Playing Sound

Homework Questions

Any questions about the homework?

Text Drawing

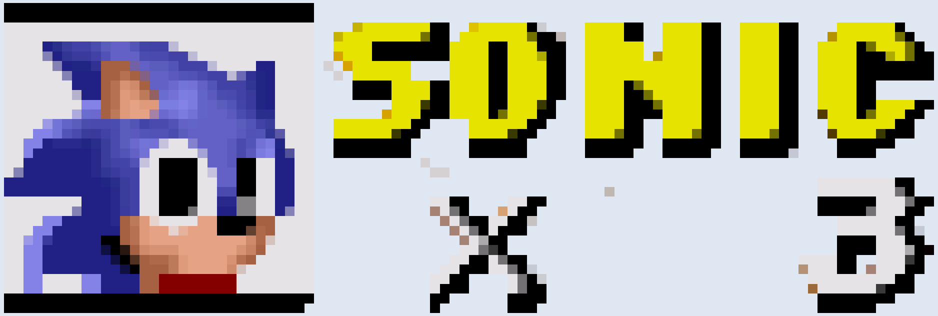
Text drawing is actually pretty simple, and just uses things you are already familiar with

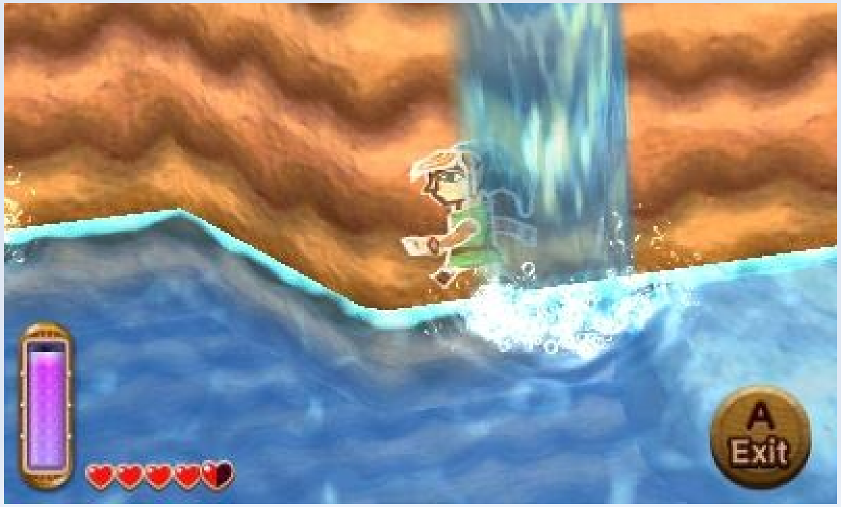
If you find a tutorial online and you don't understand it, don't use it!

Text Drawing

- Often, you don't need to actually support full text drawing.
- Logos, HUD, etc. can be plain old textures
  - Here, Last of Us logo, the word “Exit”, “Sonic” can be drawn with glDrawSprite()

THE
LAST
OF US





Text Drawing

However, for other situations...

Text Drawing

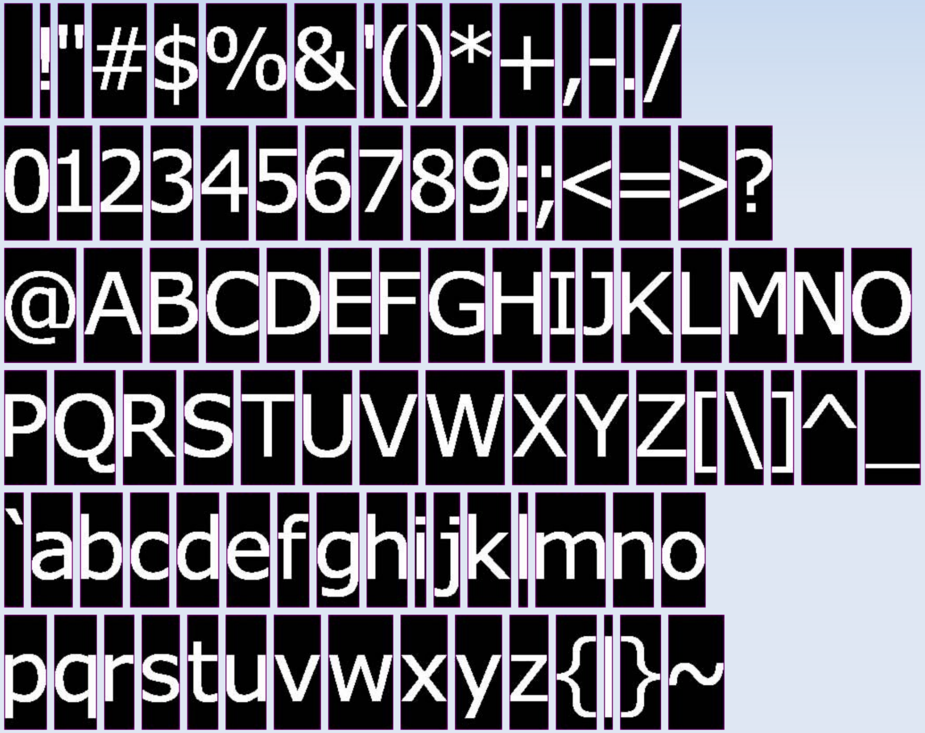
- Most APIs provide a dedicated text drawing routine.
  - Java: Graphics.drawString(str, x, y)
Graphics.setFont(font)
  - Win32: DrawText(dc, str, count, rect, align)
- Commonly, you need a “font”, a “string”, and a position

Text Drawing

- How is text represented?
- String
  - An array of bytes, that encode a sequence of codepoints, commonly using ASCII, UTF-8, or UTF-16
- Codepoint
  - A number that represents a character.
    - s 115
    - ☃ 9731

Text Drawing

- How are fonts represented?
- Font
  - A collection of codepoint → glyph mappings
- Glyph
  - A picture, in typography speak. Usually represented as a texture



Text Drawing

- Text drawing APIs usually provide the following two functions at the lowest level:
  - int DrawText(font, str, x, y)
    - Returns width of the text
    - Can only draw a single line!
  - int MeasureText(font, str)
    - Same as DrawText, but does not do any drawing
- How are these implemented?

Text Drawing

```
int DrawText(font, str, x, y)
{
    foreach (codept in str ) {
        glyph = font[codept];
        DrawGlyph(glyph, x, y);
        x += GlyphWidth( glyph );
    }
}
```

Text Drawing

- What do the data structures look like?
- class Font
- class Glyph

Text Drawing

- What do the data/def structures look like?

- class FontDef
  - int lineHeight
  - Hashtable / Array glyph def
- class GlyphDef
  - string texName
      - Becomes an int glTexture for data
  - int width

Text Drawing

Write your own glDrawText() function, based on the previous slides.

Text Drawing

- Making font textures sounds very tedious, right?
    - 96 different textures to create for just ASCII support!
- There are many tools that can help you out there
    - FontBuilder is pretty good, easy to use. Spits out a single texture, but you can cut that up in Gimp.
    - https://github.com/andryblack/fontbuilder

Text Drawing

<Example using FontBuilder>

Text Drawing

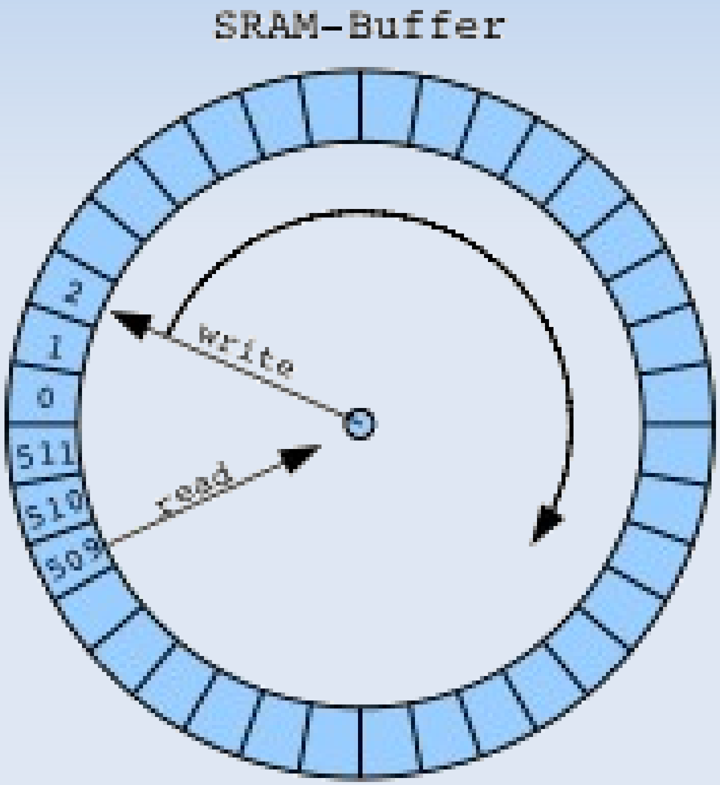
Questions?

Sound Hardware

- Not much is done in hardware (unlike Graphics)

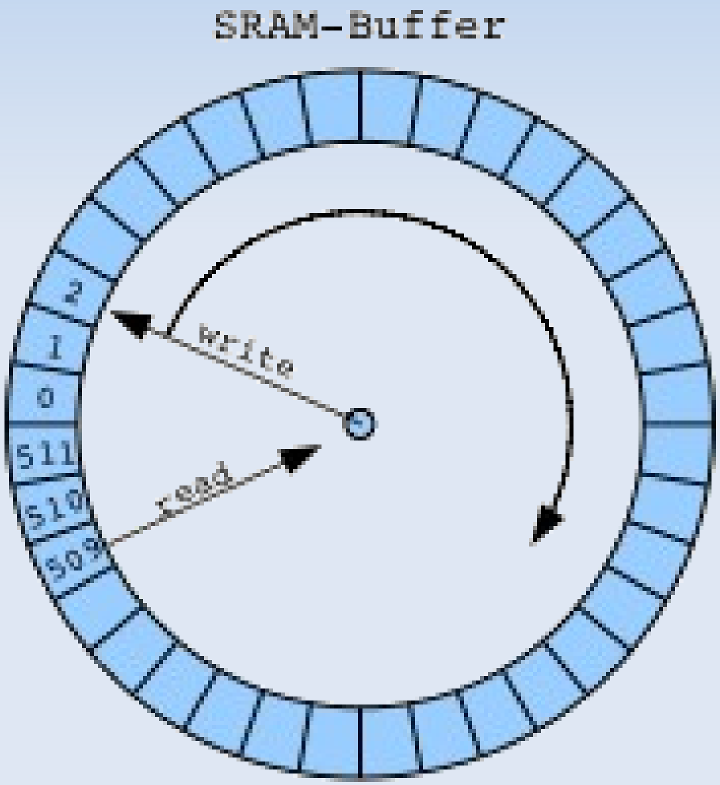
- Decompression
- Mixing
- 3D DSPs ("shaders")

- Hasn't really changed for over 10 years.



Sound Hardware

- HW has dedicated memory block it plays audio from.
- A “Ring Buffer”
- You must write the memory to the buffer before the HW reads from it.



Sound Features

- 2D / 3D Sounds
- A "2D" sound will play what was recorded exactly.
- A "3D" sound will play as if it was playing in a specific position.

- Sounds / Streams
- A "sound" is always fully uncompressed in memory
- A "stream" is loaded into memory bit by bit, as needed

Sound Features

- 2D / 3D Sounds

- Use 2D sounds unless you want positioning to be dynamic

- Sounds / Streams

- Use “sounds” for short bursts of audio

  (< 10 seconds)

- Use “streams” for background music

  (> 10 seconds)

Sound Features

- DSPs are like shaders for sound
- Echos, Pitch Shifting, Reverb, Volume fadeoff, etc.

Using FMOD

- FMOD does most of the heavy lifting for you!
  http://fmod.org

- FMOD_System_Create()
- FMOD_System_Init()
- FMOD_System_CreateSound()
- FMOD_System_CreateStream()
- FMOD_System_PlaySound()
- FMOD_Channel_SetLoopCount()

Using FMOD – Initialization

- FMOD_System_Create()
  - Load the FMOD library
- FMOD_System_Init()
  - Initalize the FMOD library

```
FMOD_SYSTEM* fmod;
// Load jump sound effect
FMOD_System_Create(&fmod);

// Initalize FMOD with up to 100 sounds playing at once
FMOD_System_Init(fmod, 100, FMOD_INIT_NORMAL, 0);
```

Using FMOD – Loading Audio

- FMOD_System_CreateSound()
  - Load audio from disk as a sound (fully decompressed into RAM)
- FMOD_System_CreateStream()
  - Load audio from disk as a stream

```
FMOD_SOUND* jump;
FMOD_SOUND* bgMusic;
FMOD_System_CreateSound(
    fmod, "mariojump.mp3", FMOD_DEFAULT, 0, &jump);
FMOD_System_CreateStream(
    fmod, “level1_bg.mp3”, FMOD_DEFAULT, 0, &bgMusic);
```

Using FMOD – Playing Audio

- FMOD_System_PlaySound()
  - Play a specified sound until it finishes.
  - Optionally can take a “channel” to play on, replacing the currently playing thing

```
// Play a “jump” sound
FMOD_System_PlaySound(fmod, jump, 0, false, NULL);

// Change the background music
FMOD_System_PlaySound(fmod, bgMusic, 0, false, &bgChan);
```

Using FMOD – Playing Audio

- How did we get the bg channel?

```
FMOD_CHANNEL* bgChan;

// Play nothing on a BG channel, start it paused
FMOD_System_PlaySound(
    fmod, NULL, NULL, true, &bgChan);
```

Using FMOD – Looping

- FMOD_Channel_SetLoopCount()
  - Set how often this channel should loop

```
FMOD_CHANNEL* bgChan;

// Loop this channel forever.
FMOD_Channel_SetLoopCount(bgChan, -1);
```

Using FMOD

- One more function
- FMOD_System_Update()
  - Update the FMOD runtime in prep for a new graphics tick

```
// Call this once per graphics frame
FMOD_System_Update(fmod);
```

Best Practices

- Initialize FMOD and a background channel at startup.
- Load sounds and streams at level start.
- Play sounds when appropriate

- Error checking!

Java Equivalent

- I've created a Java equivalent using javax.sound.sampled
- Load via Sound.loadFromFile
    - No distinction between streaming and sounds
- Play via sound.play / sound.playLooping
- Stop via clip.stop / clip.close

Audio

Questions?

Final Homework

- Add a HUD to your game

- Some things to add to your game (choose two):
  - Score count
  - Life count
  - Enemies remaining