Task 18 Spike: Game Resource Management (Loading)

CORE

Context

Playing sounds on demand for a game, based on game events, and playing background music, are key components to creating entertaining and immersive game environments. The facility to load 2D images from file and display them to screen is a critical part of many software applications, especially 2D and 3D games. Using a library to support this functionality requires the developer to understand a libraries API and 2D graphics terminology.

Knowledge/Skill Gap:

The developer needs to know how to use a framework to load, play and control game sound and music.
The developer needs to know how to load and display multiple images, including sub-regions of one image onto another.

Goals

For this case we will use SDL2 to provide sound and music playback and keyboard event response support. We will use SDL2 for presenting images and sub-regions of regions.

Create a simple application, using SDL2, that demonstrates the following features.

1. Keys 1, 2, and 3 will each play a unique sample sound as soon as each key is pressed even if that sound is already playing.
2. Play or pause (not stop) background music in response to key-down press “0” (zero) being used as a toggle.

Create a graphical 2D application capable of displaying images. Your application must:

1. Display a single image as the background image for your application, which can be toggled “on” or “off” using the “0” (zero) key
2. Load one other image that contains three identifiable sub regions (tiles) within it
3. Define three rectangles that specify the sub-region (“part”) for each tiles image
    a. Display each tiles image to a unique random location using a toggle “on” or “off” in response to the 1, 2 and 3 number keys

Expected Output

Repository

1. Code
2. Spike Report

Canvas

1. Spike Report

Notes

- Find and read documentation and tutorials related to simple (not complex or extended) image loading and display in your framework. Note – keep this as simple as possible.
- Create two of your own simple images (but do not waste time on this) saved as simple format. (.bmp is enough - there's no need to launch into more complex formats)
- Make sure you are aware of the bit-depth of your images and the screen. Always “optimize” your images to the current screen bit-depth to avoid performance penalties.

- Strongly suggest that you display messages to the console that describe what is happening and help debug your program -- such as “loading image1.bmp”, “tile 3 display ON at location (10, 40)” and so on.
- Do not over-engineer this; if you are implementing classes or using a number of libraries you have almost certainly gone too far! Just because you can does not mean you should…
- Find and read tutorials for playing a sound when an event occurs, and for playing and pausing the playback of music. (Note – you need to PAUSE the music, not just stop and start it again from the start.)
- Create or download some sounds and a music file suitable for your intended work. (Ensure you have the right licenses for any sound or music.)
- Your keyboard input spike will give you suitable code starting point for response to key events. Keep it simple.
- You may need to add debug code to your work to ensure systems are initialising and loading as needed. For example, if you are unable to load a file (file not found?) make sure *you* know about it!
- You will need to research an appropriate format for your sound file.