Class

MAFlashingLightsProcessor

A class that processes a framebuffer object to detect and dim sequences of flashing lights.

iOS 17.0+ | iPadOS 17.0+ | Mac Catalyst 17.0+ | macOS 14.0+ | tvOS 17.0+ | visionOS 1.0+

```
class MAFlashingLightsProcessor
```

Overview

A device with the Dim Flashing Lights setting on automatically dims the brightness of flashing effect sequences when it detects them in video content. If your app performs custom video drawing instead of using AVFoundation APIs, you can use the `MAFlashingLightsProcessor` class to detect and mitigate sequences of flashing effects in your video content.

The following example shows how you might incorporate `MAFlashingLightsProcessor` into code that uses Core Video APIs.

```
import MediaAccessibility
import CoreVideo
import OSLog

private let processor = MAFlashingLightsProcessor()
private let logger = Logger()

func readVideoBuffer() {

    // Confirm that the Dim Flashing Lights setting is on before processing video.
    if !MADimFlashingLightsEnabled() { return }
```

```
// Retrieve the CVPixelBuffer from your video content.
// ...

// Get the IOSurface that backs the pixel buffer.
guard let inSurface = CVPixelBufferGetIOSurface(pixelBuffer)?.takeUnretainedValu
    logger.debug("Can't initialize input surface.")
    return
}

// Use the properties of the pixel buffer to initialize an output IOSurface.
guard var outSurface = IOSurface(properties: [
    .width: CVPixelBufferGetWidth(pixelBuffer),
    .height: CVPixelBufferGetHeight(pixelBuffer),
    .bytesPerRow: CVPixelBufferGetWidth(pixelBuffer) * 4,
    .bytesPerElement: 4,
    .pixelFormat: CVPixelBufferGetPixelFormatType(pixelBuffer)
]) as IOSurfaceRef? else {
    logger.debug("Can't initialize output surface.")
    return
}

// Verify that the input IOSurface is compatible with the flashing lights proces
if processor.canProcessSurface(inSurface) {

    // Analyze input IOSurface for flashing light sequences
    // and write mitigated content to output IOSurface.
    let result = processor.processSurface(inSurface, outSurface: &outSurface,
                                          timestamp: CFAbsoluteTimeGetCurrent()

    if result.surfaceProcessed {
        logger.debug("""
        Processed content with flashing lights intensity \(result.intensityLevel
        and mitigated output with mitigation level \(result.mitigationLevel).
        """)

        // Convert the mitigated output surface back to CVPixelBuffer
        // and draw the video content.
        // ...

    } else {
        logger.debug("Can't process input surface.")
    }
}
}
```

For more information, see Flashing lights.

Topics

Checking compatibility

`func canProcessSurface(IOSurfaceRef) -> Bool`

Returns a Boolean value that indicates whether the flashing lights processor can process the content in the surface for sequences of flashing lights.

Processing video content

`func processSurface(IOSurfaceRef, outSurface: inout IOSurfaceRef, timestamp: CFAbsoluteTime, options: [MAFlashingLightsProcessor.OptionKey : Any]?) -> MAFlashingLightsProcessor.Result`

Processes a surface by analyzing pixels for sequences of flashing lights and mitigates them by dimming the content.

`struct Result`

An object that reports the result of the flashing lights processor.

`struct OptionKey`

Options for the flashing lights processor.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`

```
Equatable
Hashable
NSObjectProtocol
```

See Also

Dim flashing lights

{} Responding to changes in the flashing lights setting

Adjust your UI when a person chooses to dim flashing lights on their Apple device.

`func MADimFlashingLightsEnabled() -> Bool`

Returns a Boolean value that indicates whether the flashing lights setting is enabled on the device.

`let kMADimFlashingLightsChangedNotification: CFString`

A notification that posts when a person changes the flashing lights setting on the device.