

# ScummVM Conceptual Architecture

---

Presented by Jacob McMullen, and Reid Stobo

<https://youtu.be/i-MpPEIrYEc>

# TEAM 15:



Ryan Jacobson  
(Group Leader)



Evan  
Kreutzwiser



Reid  
Stobo



Jacob  
McMullen

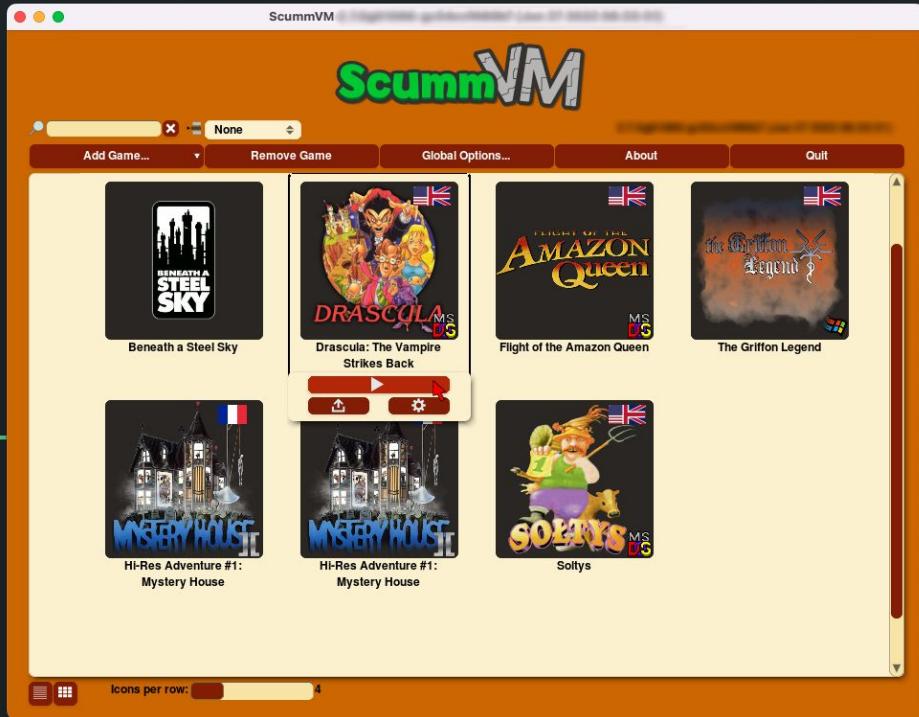


Kashan  
Rauf



Mike  
Stefan

# ScummVM



Open Source Platform for running classic computer adventure games and RPGs on systems they were never designed for

Supports Windows, Linux, IOS, and Android as well as less conventional OS like the PS Vita, Nintendo Wii, and Haiku OS.  
Supports a total of 19 Platforms

Supports over 325 classic games!

# Derivation process



Log in

Page Discussion Read View source View history Search ScummVM Q

## About

Language: English · Deutsch · français · italiano

Contents [hide]

- 1 Brief description
- 2 A more thorough description
  - 2.1 The inner workings of adventure and role-playing games
  - 2.2 The scripting language of a game
  - 2.3 Where does ScummVM fit in?

### Brief description

ScummVM is a program which allows you to run certain classic adventure and role-playing games, provided you already have their data files. The clever part about this: ScummVM just replaces the executables shipped with the games, allowing you to play them on systems for which they were never designed!

For more information about the makers, you may read the [ScummVM History](#) or the [Developers Bios](#).

Home  
Supported platforms  
User documentation  
  
Games  
Get the games  
Supported games  
Unsupported games  
International game titles  
Game engines  
Reporting unknown game variants  
Submitting screenshots  
  
Developers  
Developer Central  
Compiling ScummVM  
Coding conventions  
Code formatting

The official wiki has developer documentation about the system.

# Derivation process

**Personal blogs and presentations by the developers made for excellent resources for outsiders new to ScummVM.**

## Portability

- ScummVM architecture evolved early on to make it easy to abstract backend and common functions ('Osystem' class supplying a Osystem backend class (with platform subclassing where necessary))  


```
graph LR; Backend[Osystem Backend<br>(Platform Specific)] <--> Common[Osystem Common]
```
- Carefully developed coding standards to encourage a common denominator C++ implementations. No STL, ETC...  
[http://wiki.scummvm.org/index.php/Coding\\_Conventions](http://wiki.scummvm.org/index.php/Coding_Conventions)
- Ssennaidne
- Segment size limits on various platforms

LINUX.CONF.AU  
THE FUTURE OF OPEN SOURCE  
2011 HOBART



# ScummVM

"ME HOO!"

Source: [ScummVM news](#)

to come off the bench for  
play again on ScummVM  
to mention **the last**  
**engine**). Play with the  
your own! This game also  
stars Kevin Garnett and

franchise, controls are  
load and keyboard options.  
3 on 3 basketball. Don't  
nut for a *slamma jamma!*  
from Sunny Day  
ers on your way to a

old CD and fire up the

People (43)

- Criezy's Blog
- Dreammaster's disassembly blog
- Dylan Servilla - ScummVM Blog
- Gobsmacked
- Google Summer of Code – Vyzigold
- Google Summer of Code with ScummVM
- grisentl's GSoC blog
- Groovie, baby
- GSoC 2020 ResidualVM
- GSoC 2021 – ayyg
- GSoC 2021 – av\_dx
- GSoC 2021 – djsrv

# Derivation process



navigation

- Main page
- SCI Documentation
- The SCI Parser

## Main Page

### Welcome to the SCI Programmers Wiki.

A Wiki for the SCI Programming Community. SCI is the game engine created by Jeff Stephenson for Sierra On-Line, which was used between the years of 1989 to 1997. It superseded Sierra's earlier AGI engine. It was used to create such games as *King's Quest IV* to VII, *Space Quest III* to 6, *Quest for Glory I* to IV and *Gabriel Knight I* and II. With the availability of such tools as Brian Provinciano's SCI Studio 3 and Phil Fortier's SCI Companion, you can create your own games based on the same engine that Sierra used. This Wiki is a knowledgebase and collection of resources for those working with SCI. Additional help can always be found on the SCI Programming Community Message Board. For help with Sierra's games, see The Sierra Help Pages.

log in

Documentation about SCI engine's  
implementation from the original FreeSCI  
project no longer exists; the official SCI wiki at  
times provided insight.

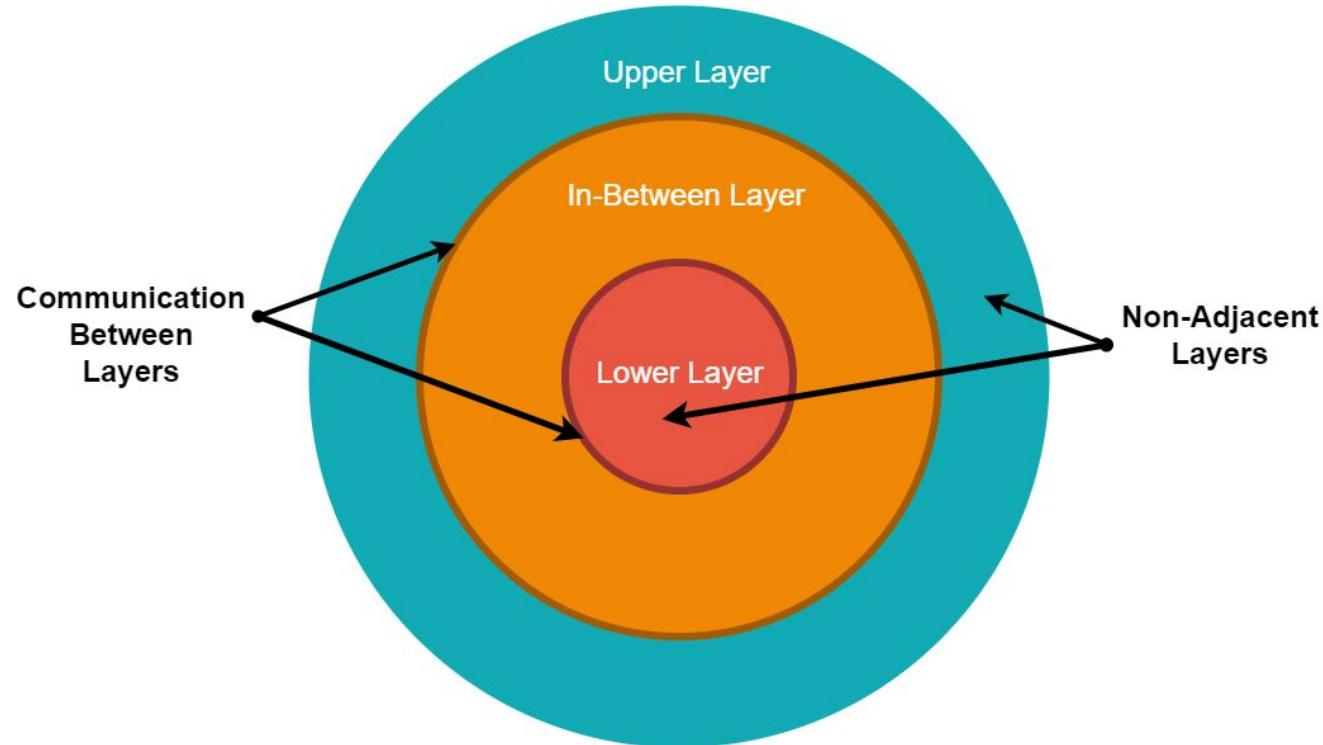
# Derivation process

The GitHub repository page for `scummvm/scummvm` (Public). The sidebar on the left shows the master branch, 40 branches, and 130 tags. The main content area displays a list of recent commits from user `athrxx`, including fixes for Japanese text, OpenGL reenablement, license file updates, and various backend and common module changes. The right sidebar provides an 'About' section with links to the project website (`www.scummvm.org`), multiplatform support, RPG engine, adventure engine, and adventure games. It also lists the README, GPL-3.0 license, activity, custom properties, star count (2.3k), watching count (136), fork count (1k), and a report repository link. The 'Releases' section shows the latest release, ScummVM 2.8.1: "Oh MMy!" (Latest), released on Mar 31.

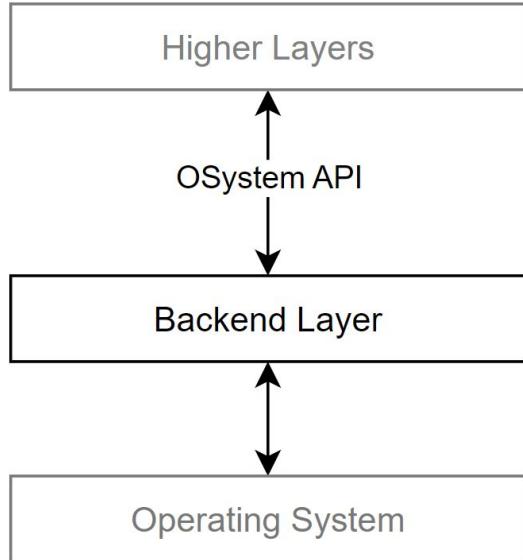
Commit	Message	Time Ago
<code>1fefea0</code>	SCI: (QFG/PC-98) - fix Japanese text ...	15 minutes ago
<code>...</code>	CREATE_PROJECT: Reenable OpenGL on a...	2 months ago
<code>...</code>	LICENSES: Make sure every license file has...	5 months ago
<code>...</code>	BACKENDS: Remove leftover GP2X referen...	2 weeks ago
<code>...</code>	BACKENDS: moved imgui_logger code fro...	5 hours ago
<code>...</code>	JANITORIAL: Fix typo in main	2 weeks ago
<code>...</code>	COMMON: fixed whitespaces	4 hours ago
<code>...</code>	BACKENDS: Remove leftover GP2X referen...	2 weeks ago
<code>...</code>	SAILFISH: New port for SailfishOS	4 days ago
<code>...</code>	DOCS: Update iOS Apple Pencil controls	last week
<code>...</code>	SCI: (QFG/PC-98) - fix Japanese text color...	15 minutes ago
<code>...</code>	GRAPHICS: Simplify the generic blending r...	2 weeks ago

The GitHub repository contains developer resources and changelogs, we avoided looking at source code to not confuse conceptual/concrete.

# Layered Architecture



# Backend Layer



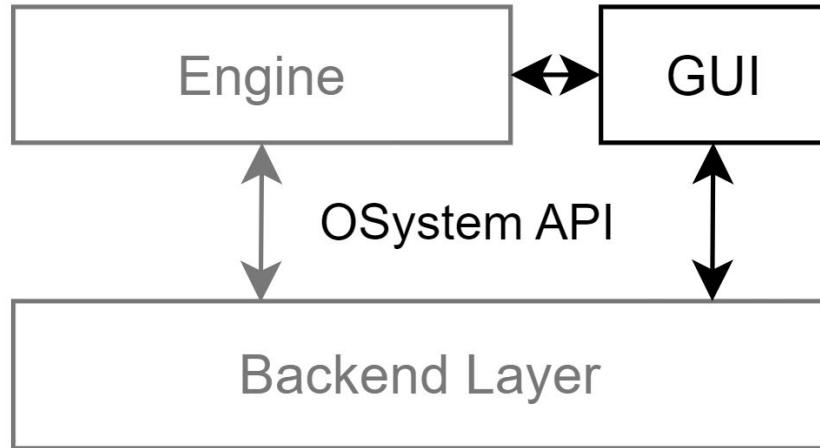
Legend:

Conceptual  
Component

—Communication→

- Platform-specific
- Communication with host OS
- Defines OSystem API to handle OS requests
- Handles audio & video output, filesystem access, user input

# Launcher GUI



Legend:

Conceptual  
Component

—Communication→

- Game launcher
- Game options management
- Display and audio settings
- Save state management

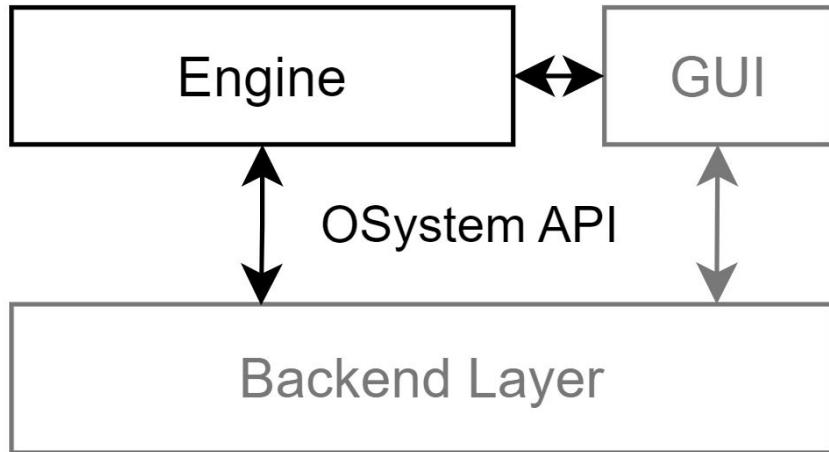
Game Launcher



Global Options  
Menu



# Engine



Legend:

Conceptual Component

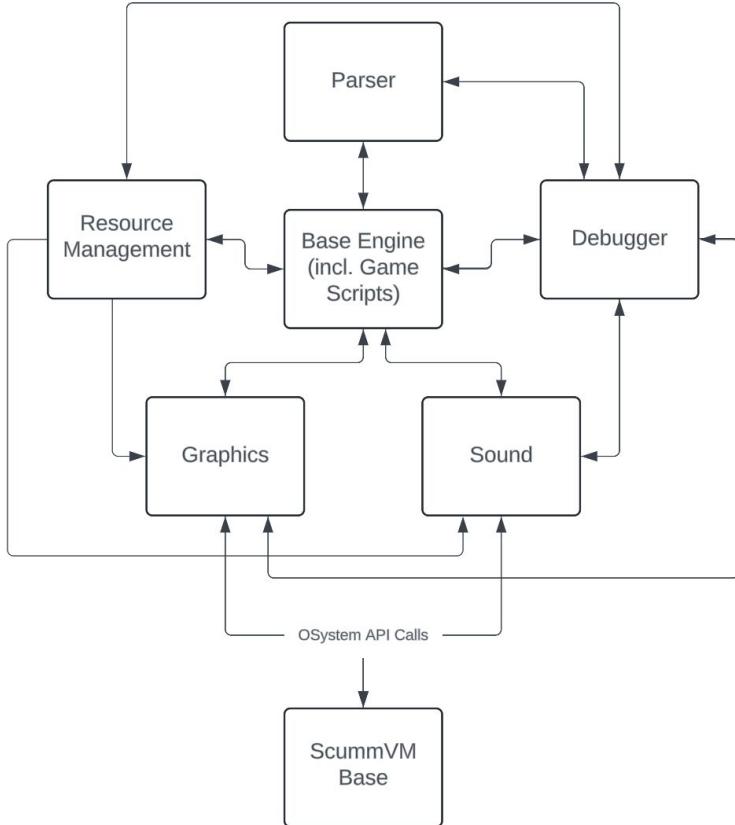
—Communication→

- Takes the place of original game's executables
- Script interpreters, resource management, rendering, etc.
- Loads files of original game
- Uses OSSystem API to receive input, use filesystem, play audio, and display video.

Communication between GUI and MetaEngine:

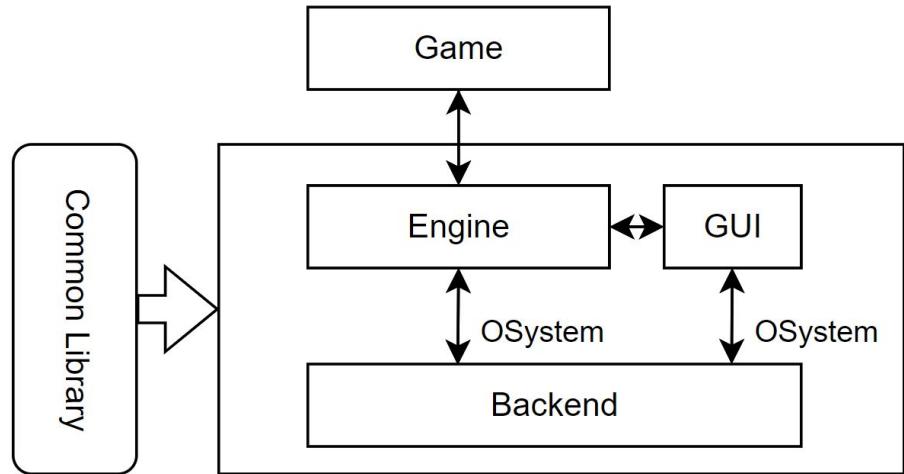


# SCI Engine



- Uses an Object-Oriented Architectural Style
- Subsystem singletons divide responsibility
- Graphics and Sound subsystems interact with the OSSystem API

# Common Library



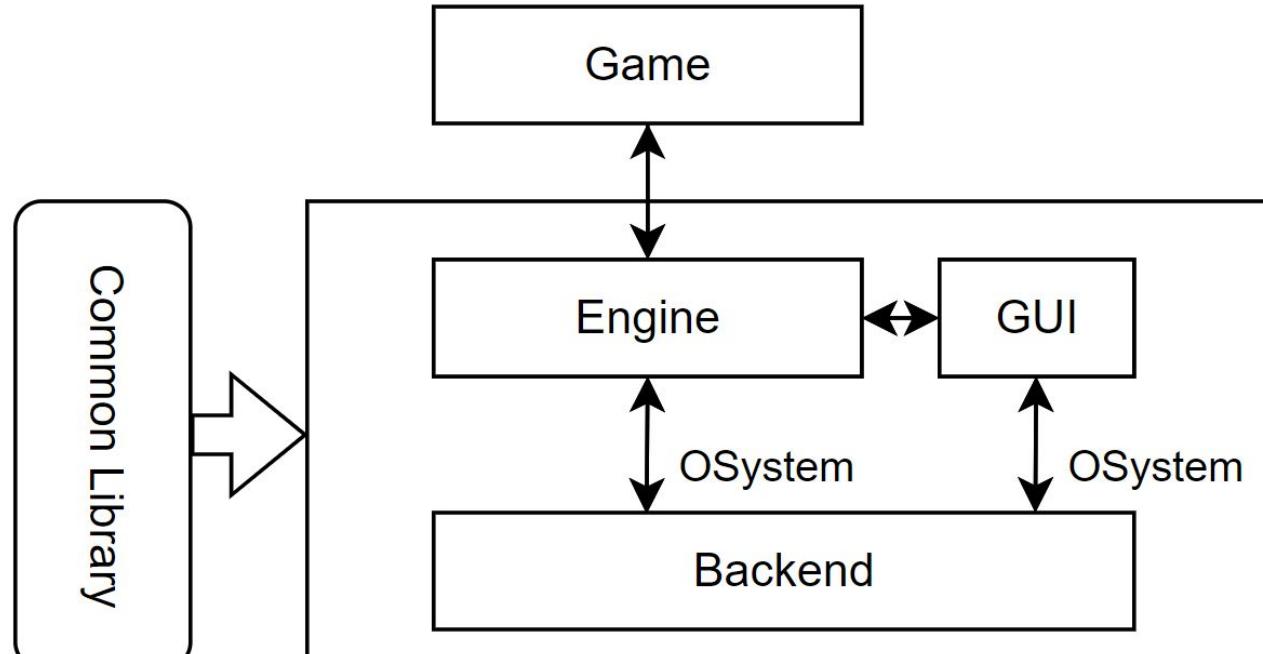
Legend:

Conceptual  
Component

—Communication→

- Takes the place of original game's executables
- Script interpreters, resource management, rendering, etc.
- Loads files of original game
- Uses OSystem API to receive input, use filesystem, play audio, and display video.

# ScummVM Architecture



Legend:

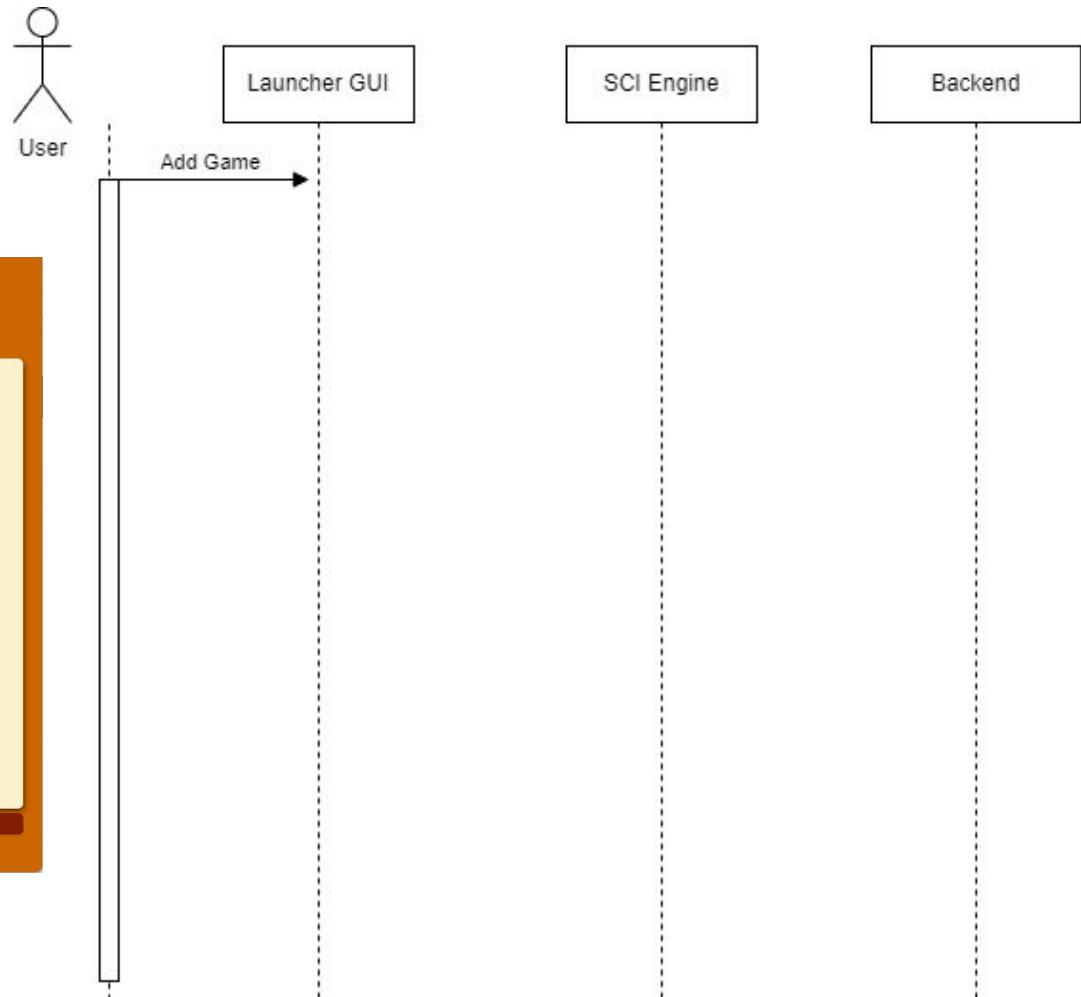
Conceptual  
Component

—Communication→

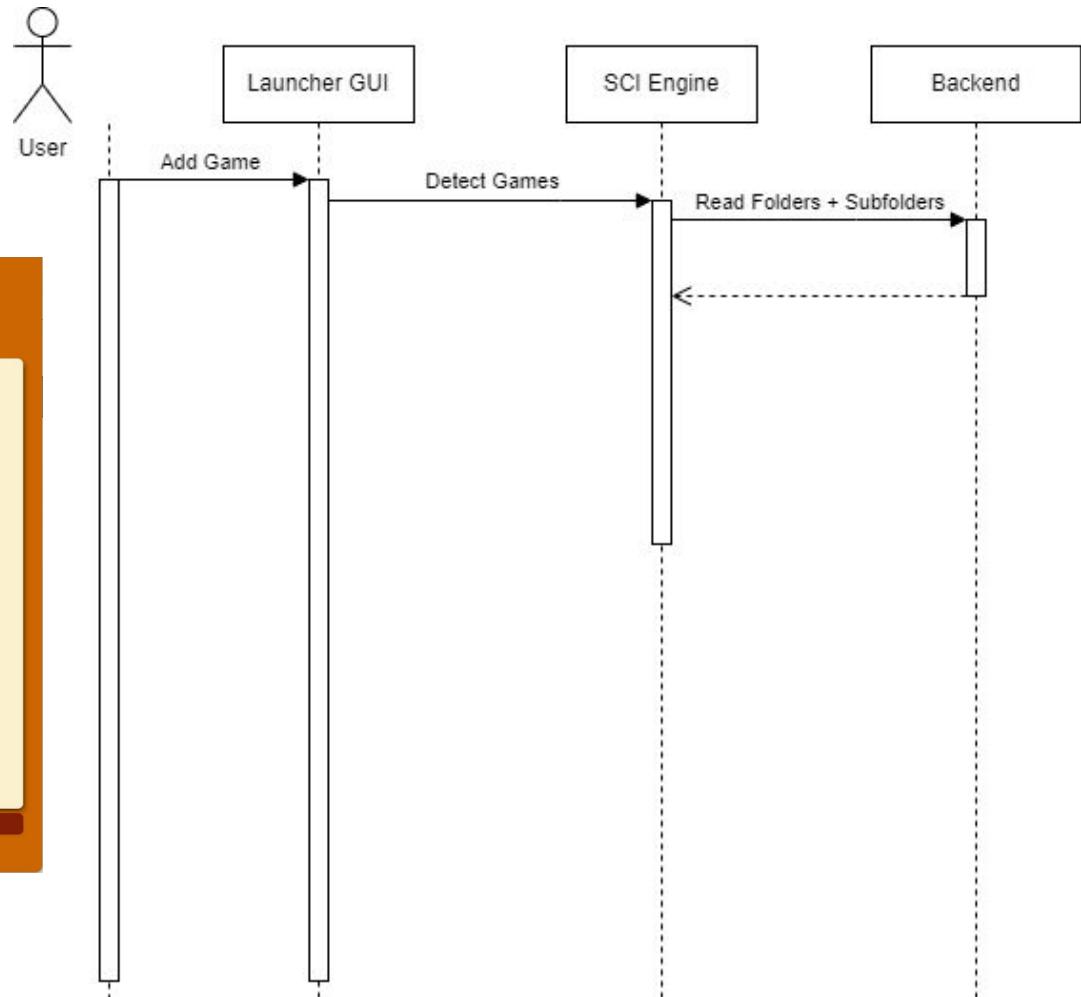
# Concurrency

- Primarily single threaded
- Backends and audio may introduce more threads
  - Realtime audio
  - Timers
  - Event queues
- Engines use cooperative multithreading inside interpreters
- Additional backend threads introduce asynchronous behavior
  - Otherwise, engines & gui completely single threaded
- Not all target platforms make use of threading

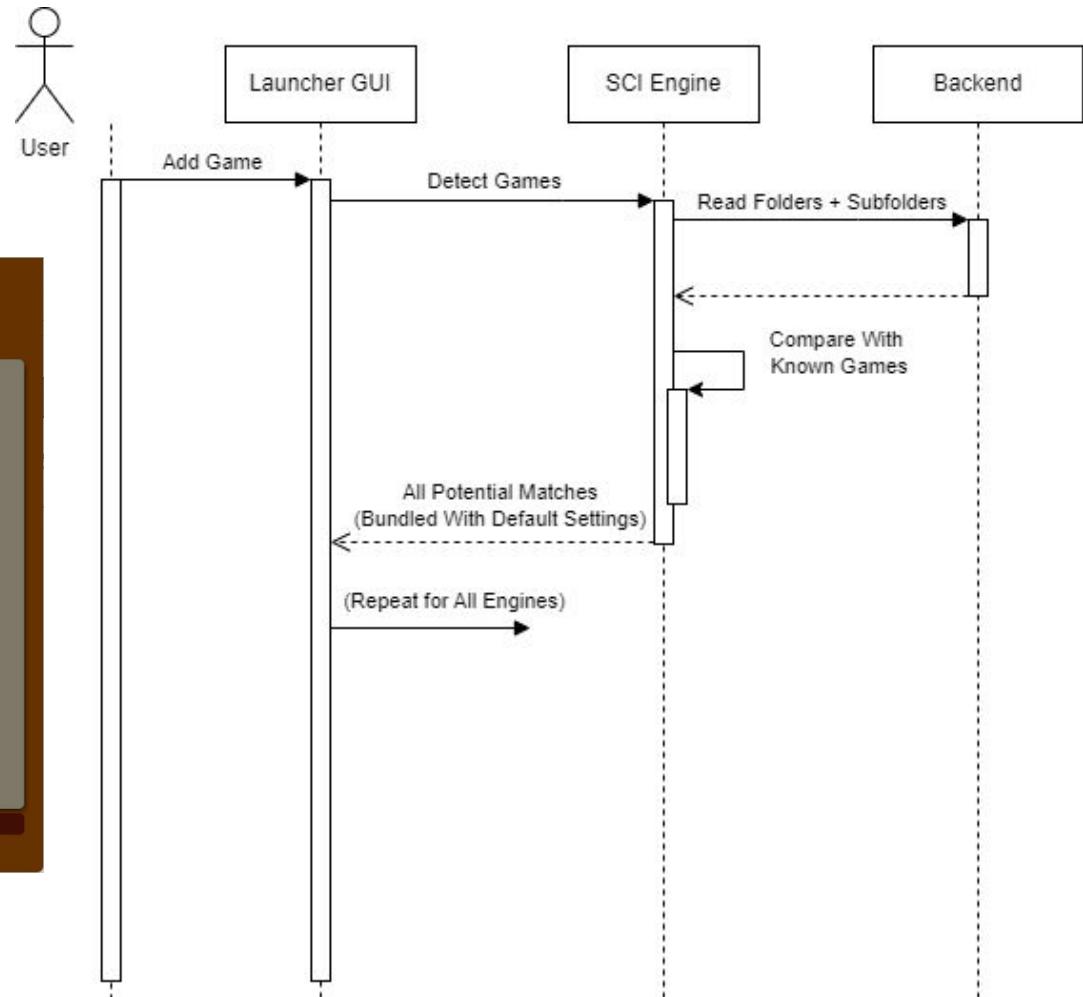
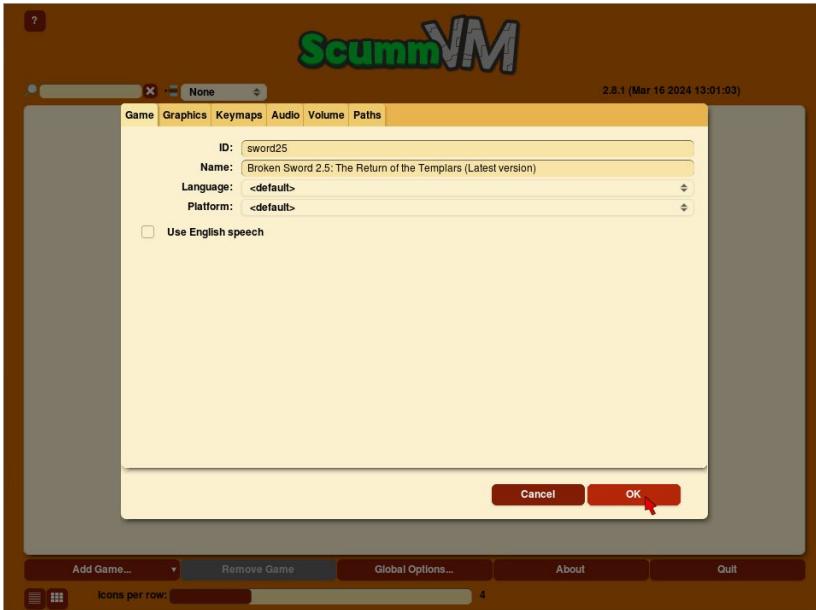
## Use Case 1: Recognizing Games



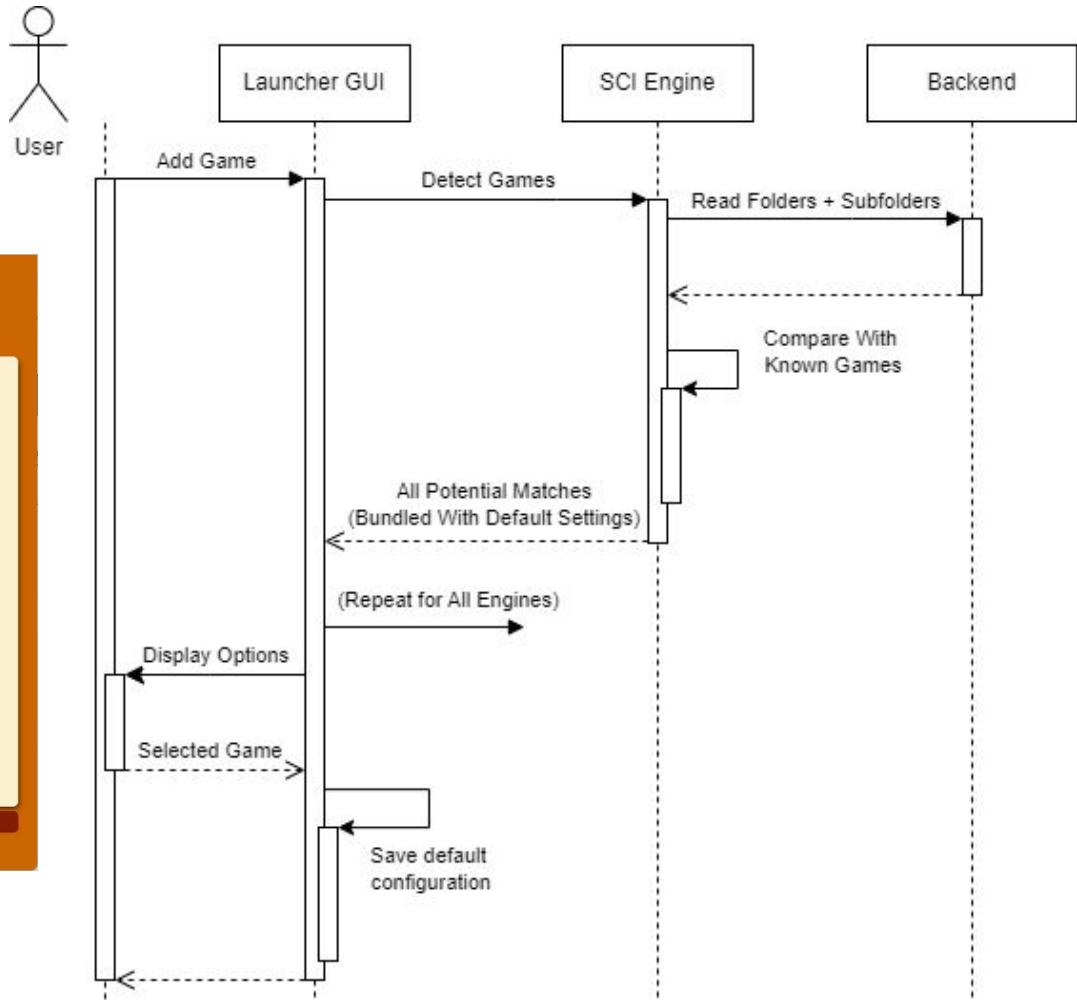
# Use Case 1: Recognizing Games



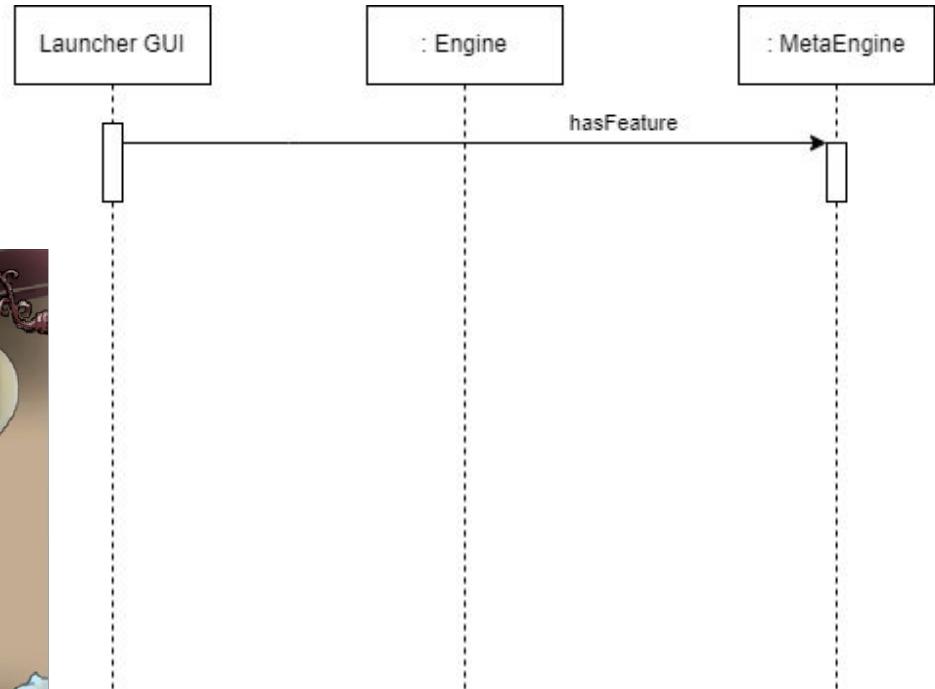
# Use Case 1: Recognizing Games



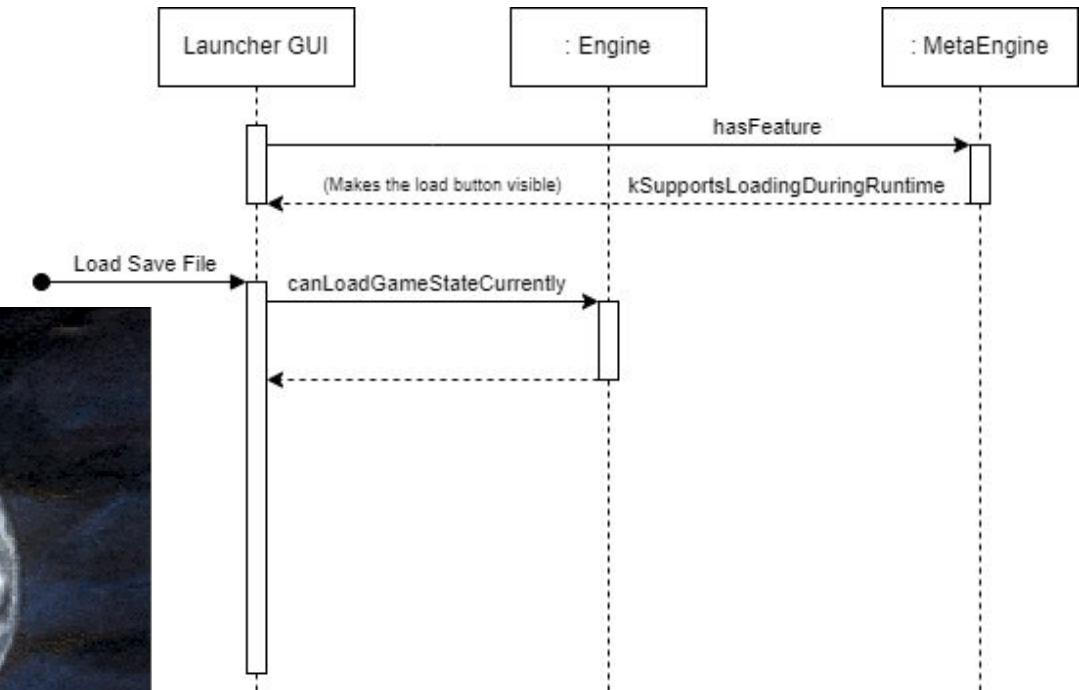
# Use Case 1: Recognizing Games



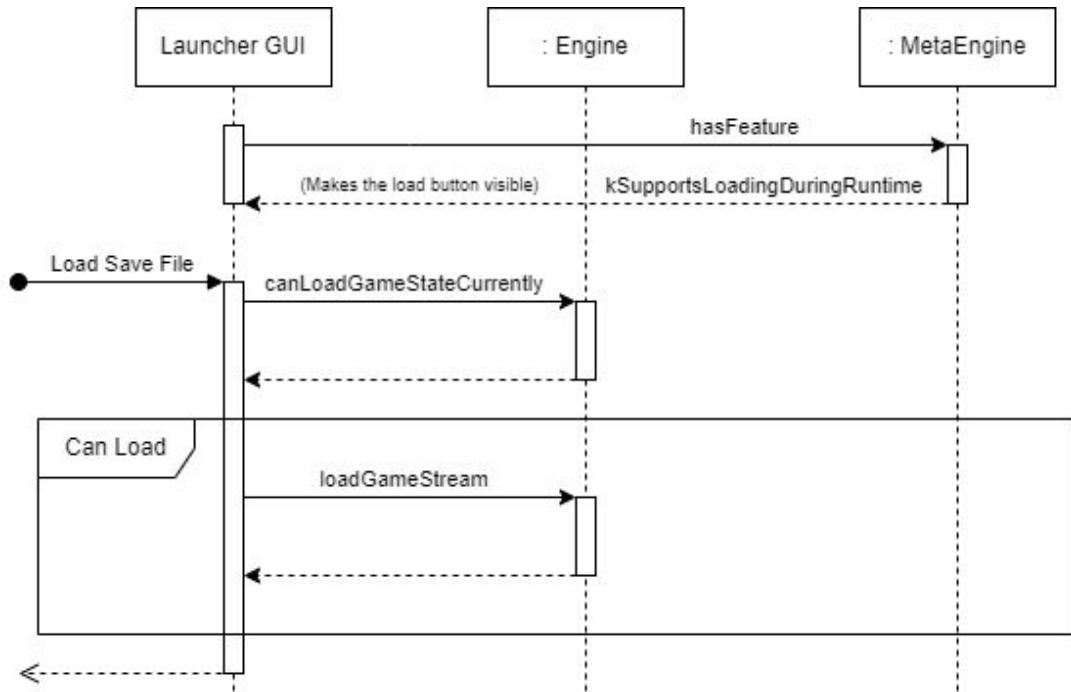
# Use Case 2: Loading Save Files During Runtime



## Use Case 2: Loading Save Files During Runtime



## Use Case 2: Loading Save Files During Runtime



# Closing Thoughts

---

