Architectural Enhancement of the ScummVM Engine

Presentation URL: https://youtu.be/mbqtl4A5_TE

Group 18

Harry George (Leader) - Architectural Impact, Testing and Validation, Sequence Diagrams

Denis (Presenter) - Architectural Impact, Alternatives for Realization, Conclusion, Presenter

Kurtis Marinos (Presenter) - SAAM Analysis, Risks and Mitigation, Presenter

Ben Hilderman - Sequence Diagrams, Interactions with current system

Riley Spavor - Proposed Enhancement, Abstract

Jacob Skiba - Architectural Impact, Alternatives for Realization

Overview

Proposed Enhancement: Introduce achievements and leaderboards to ScummVM.

Features:

- **Badges, points, and titles for milestones.**
- **Dynamic leaderboards with filters (region, game type, speed).**
- **Player profiles to showcase progress.**

Impact:

- **Boost engagement and replayability.**
- **Foster community and social interactions.**
- **Bridge nostalgia with modern expectations.**

Benefits and Motivation

Current Limitations:

- **No progress tracking**
- **Lack of social connectivity**
- **Limited engagement for younger audiences**

Benefits:

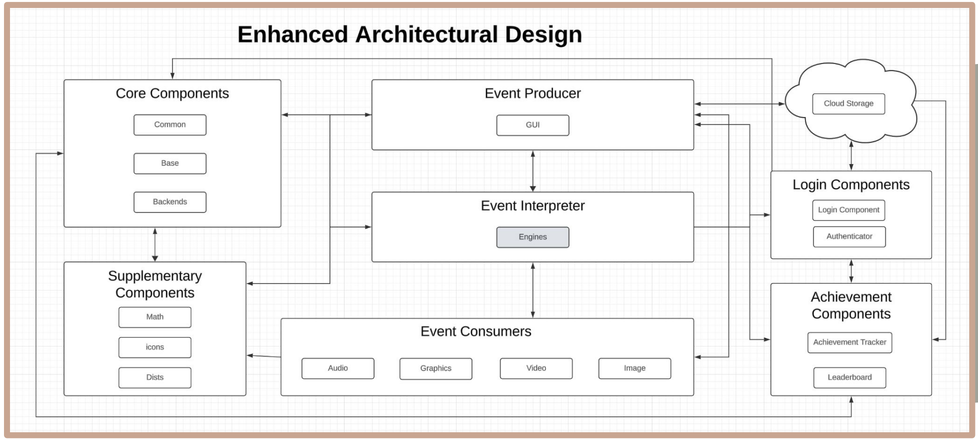
- **Boosts user engagement**
- **Fosters community interaction**
- **Bridges classic gaming with modern expectations**

System Architecture

Integrated with Existing ScummVM

Components:

- **Engines Subsystem: Event generation**
- **Common Subsystem: Data storage and tracking**
- **GUI Subsystem: Display and interaction**
- **Backend Subsystem: Leaderboard management**



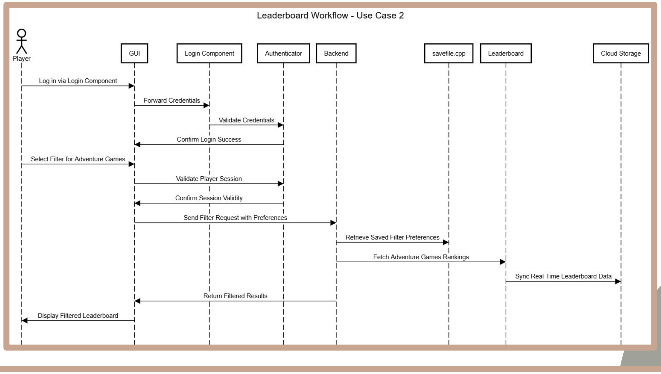
Maintains event-driven, interpreter-based architecture

Use Cases

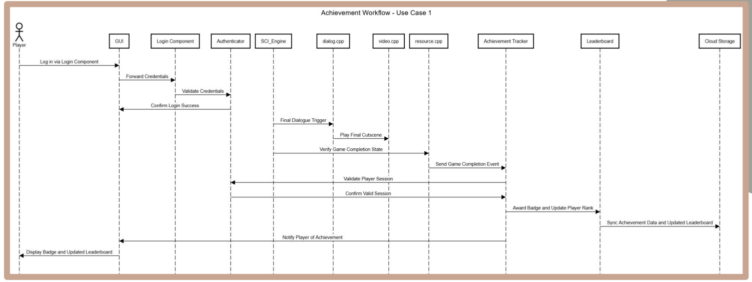
Three Primary Workflows:

1. **Achievements Workflow**
   - ■ **Game completion triggers badge award**
   - ■ **Update player rank**
   - ■ **Sync with cloud storage**
2. **Leaderboard Workflow**
   - ■ **Filter rankings**
   - ■ **Dynamic real-time updates**
3. **Login and Personalization**
   - ■ **Secure authentication**
   - ■ **Retrieve personal achievements**

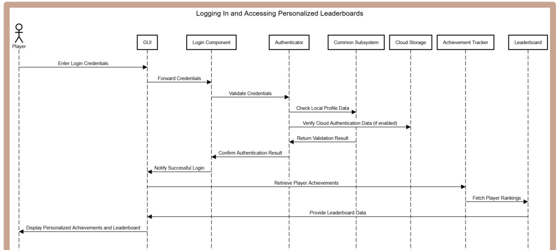
Use Case I


Leaderboard Workflow - Use Case 2
Player
GUI
Login Component
Authenticator
Backend
savefile.cpp
Leaderboard
Cloud Storage
Log in via Login Component
Forward Credentials
Validate Credentials
Confirm Login Success
Select Filter for Adventure Games
Validate Player Session
Confirm Session Validity
Send Filter Request with Preferences
Retrieve Saved Filter Preferences
Fetch Adventure Games Rankings
Sync Real-Time Leaderboard Data
Return Filtered Results
Display Filtered Leaderboard

Use Case 2



Use Case 3



Implementation Approaches

Two Potential Strategies:

1. **Incremental Integration**
   - **Gradual, low-risk implementation**
   - **Minimal system disruption**
   - **Phased development**
2. **Centralized Overhaul**
   - **Standalone module development**
   - **Faster deployment**
   - **Higher initial risk**

Recommended Approach

Incremental Integration Chosen

Rationale:

- Aligns with ScummVM's architectural principles
- Minimizes risk
- Allows iterative testing
- Ensures compatibility with diverse game engines

Risk Mitigation

Risk Mitigation

- Key Risk Areas:
  - Security
  - Performance
  - Maintainability
  - Usability
  - Privacy
  - Scalability

Mitigation Strategies:

- Server-side validation
- Asynchronous data syncing
- Modular design
- User privacy controls

Testing and Validation

Comprehensive Testing Plan:

- **Functional Testing**
  - **Achievement tracking**
  - **Leaderboard updates**
- **Integration Testing**
  - **Seamless interaction with existing subsystems**
- **Non-Functional Requirement Validation**
  - **Performance**
  - **Usability**
  - **Scalability**

Future Enhancements

Future Possibilities

- Potential Enhancements:
  - Online leaderboards
  - Additional achievement types
  - Expanded community features
- Continued focus on:
  - Game preservation
  - User engagement
  - Platform modernization

Conclusion

Achievement System Delivers:

- Modern gaming features
- Community engagement
- Respect for retro gaming heritage

Strengthens ScummVM's role in game preservation

Ensures continued relevance and user enjoyment

Thank You

References

1. ScummVM Team. (2023). *ScummVM main repository*. GitHub. Retrieved from https://github.com/scummvm/scummvm
2. ScummVM Forums. (2010). *How is ScummVM's Architecture Structured?* https://forums.scummvm.org/viewtopic.php?t=7886. Accessed 12 Nov. 2024.
3. SDL Wiki. *SDL3 FrontPage*. *n.d.*, https://wiki.libsdl.org/SDL3/FrontPage. Accessed 14 Nov. 2024.