Research Report: Developing a system for uploading games

Abstract:

This research report provides an in-depth, step-by-step guide on how to design and implement a game upload system for arcade machines running on a Linux-based operating system. The report outlines specific operational steps for each aspect of the system's development, ensuring clarity and precision.

1. Introduction

This guide offers a comprehensive and precise set of instructions for creating a game upload system for Linux-based arcade machines. The objective is to enable first-year students at Deakin University to seamlessly upload and play their games on these machines, enhancing the overall gaming experience.

2. Research Process

2.1 Preliminary Planning

Define Objectives:
Begin by explicitly defining the objectives of the system. For instance, state that the system aims to allow students to upload and play their games on arcade machines at Deakin University.
User Requirements:
Collect user requirements through surveys and interviews, focusing on the needs and preferences of first-year students who will be using the system.
Technology Stack Selection:
Select a technology stack that is compatible with Linux and suitable for game hosting. Consider technologies like LAMP (Linux, Apache, MySQL, PHP) or Node.js with a PostgreSQL database.

2.2 Investigating Game Upload Methods

2.2.1 Arcade Game Repository Integration

Create a GitHub Repository:
Begin by creating a dedicated GitHub repository specifically for hosting games and their related assets.
GitHub Actions Workflow:
Develop detailed GitHub Actions workflows to automate the game packaging and deployment process. This should include steps such as cloning the repository, packaging game assets, and transferring games to the arcade machines.

2.2.2 Arcade Machine Functionality

Script Development:
Create a custom script using a language like Python or Shell that has the capability to fetch games from the Arcade Game Repository.
Automation:
Configure the script to run at scheduled intervals to ensure that game libraries remain up to date.
Error Handling and Logging:
Implement comprehensive error handling and logging mechanisms within the script to enhance the system's reliability.

2.3 User-Friendly Application

2.3.1 Design and Development

User Interface (UI) Design:
Invest time in designing an intuitive and user-friendly UI for the game upload application. Incorporate features such as game selection, upload options, and clear progress indicators.
Linux Compatibility:
Develop the application using technologies and frameworks that are known to be compatible with Linux systems to ensure seamless integration.
Menu Integration:
Integrate the application into the arcade machine's menu system, providing easy access for users.

2.3.2 Testing Environment

Emulation Station Integration:
Design a dedicated UI element within Emulation Station that facilitates access to and management of test games.
Launch Mechanism:
Implement a mechanism within Emulation Station that allows users to effortlessly launch and manage test games.
Administrative Control:
Add a feature to easily toggle the visibility of the test game menu for administrative control and security.

3. Implementation

Development Phase:
Begin the development phase by creating the necessary scripts, GitHub Actions workflows, and the game upload application, adhering closely to the specifications defined earlier.
Testing:

Rigorously test the entire system, first in a controlled testing environment, and then on actual arcade machines to ensure functionality and reliability.

Iteration and Refinement:

Continuously refine the system based on user feedback and real-world usage observations. Make improvements to enhance user experience and system performance.

4. Conclusion

By following these meticulously detailed instructions, you can successfully design and implement a game upload system for Linux-based arcade machines. This system will greatly enhance the accessibility and enjoyment of arcade machines for Deakin University's students.

5. Future Directions

Future enhancements and developments could include:

- Integration with student profiles for personalized game management.
- Expansion of the system to support multiple arcade machines across campus.
- Implementation of robust security measures to safeguard game integrity and system stability.