THE GAMING ROOM

Draw it Or Lose It

CS 230 Project Software Design Template

Version 3.0

Table Of Contents: Table of Contents CS 230 Project Software Design

Document Revision History

Version	Data	Author	Comments
3.0	12/09/2024	Princess Sihanourath	Architectural Recommendation

Executive Summary

The mission of The Gaming Room is to adapt the existing Android-exclusive game, "Draw It or Lose It," into a web-based application that caters to a variety of platforms. The game involves multiple teams, each with several players, who participate in a series of four rounds, each lasting one minute. During each round, an image is selected from an extensive library and one team attempts to guess the mage until the time runs out. If the initial guess is not correct, the opposing team is given a 15-second window to guess. This project aims to enhance the gaming experience by making it more accessible and engaging users across various platforms.

Requirements

- The game must have the ability to have one or more teams involved.
- Each team must have multiple players assigned to it.
- Game and team names must be unique to allow users to check whether a name is in use when

choosing a team name.

- Only one instance of the game can exist in memory at any given time.
- The game must consist of four rounds of play lasting one minute each.
- The game needs to be developed as a web-based application to serve multiple platforms.

Design Constraints

- With multiple teams playing at the same time, the game should be able to handle concurrent

requests at one time.

- The game should be able to handle an increasing number of users without lowering

performance.

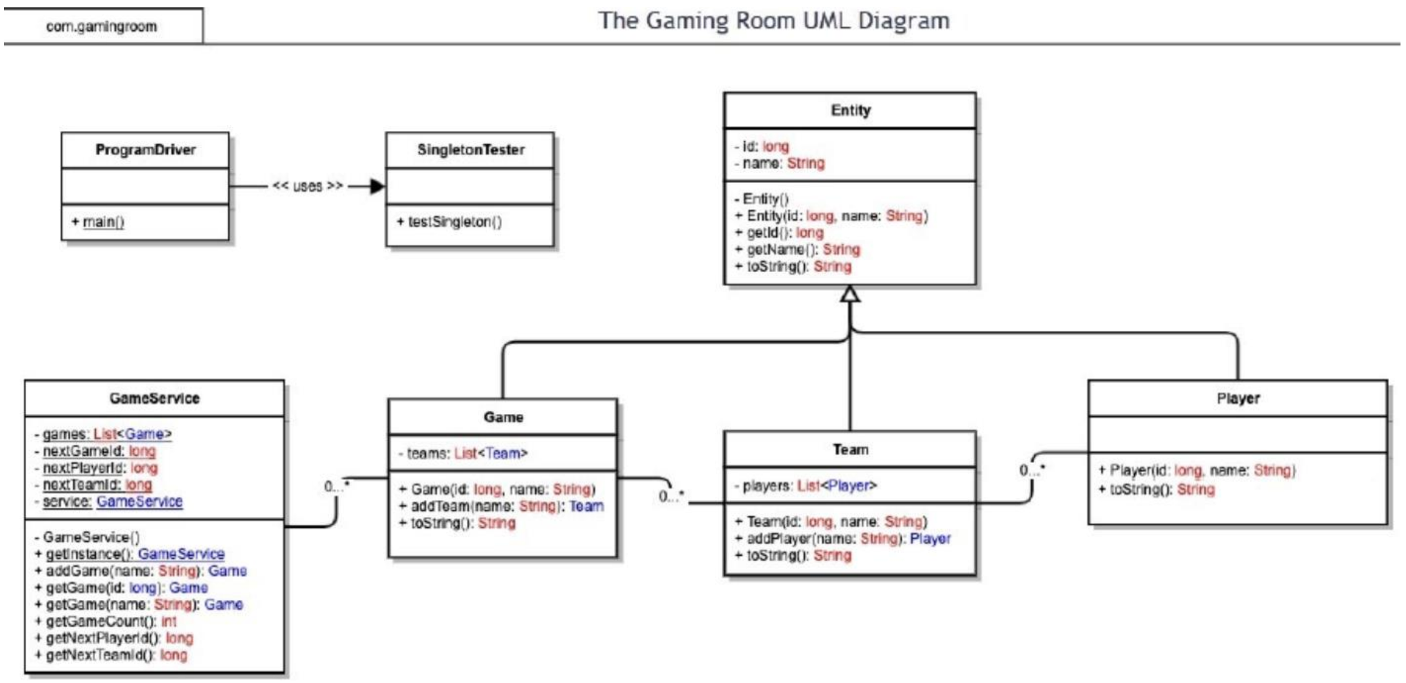
- Ensuring the security of user data and transactions is crucial in a web-based setting.
- Appropriate Security measures should be put in place to safeguard users.
- The game needs to be compatible with various platforms and browsers.
- The game should be easy to use and engaging for the end-users.

System Architecture View

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

Domain Model

The UML diagram provides six classes including ProgramDriver, SingletonTester, Entity, GameService, Game, Team, and Player. These classes demonstrate several principles of object-oriented programming. The Game and Team classes are subclasses of the Entity class, representing the principle of inheritance. This principle promotes code reusability as these subclasses can access the Entity class's attributes and methods. The diagram also shows the principle of aggregation, or the "HAS-A" relationship. This is seen where the GameService class has multiple Game objects, the Game class has multiple Team objects, and the Team class has multiple Player objects. The GameService class seems to implement the Singleton pattern, indicated by its self-referencing attribute. This pattern ensures that only one instance of the class exists in the application. Lastly, encapsulation is demonstrated where each class has private attributes and public methods. This principle hides the internal states of a class and exposes only necessary operations. These principles together contribute to efficient and maintainable software
Design.



Evaluation

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s Requirements, and look at the situation holistically, as it all must work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

Development Requirements	Mac	Linux	Windows	Mobile Devie
Server Side	Mac offers Mac OS X Server for use. According to Apple's website, Mac OS X Server costs only $20, making it a cost-effective option. However, Mac is less commonly used than Linux or Windows for these tasks.	Linux is noteworthy due to its various distributions with server capabilities. Linux Server is affordable and open-source, offering numerous resources. However, since many users are not familiar with Linux, you'll need someone with expertise to manage the server.	Cost would be like a Windows setup, as these operating systems are not open source. Time would depend on expertise, as someone who has experience with Windows would need less time and someone who does not have as much experience with Windows would need more time	Cost would not be too much of an issue with Mobile devices. Experience may not be too much of an issue, as mobile devices can be easier to work with. More time would be needed, as there are multiple operating systems and multiple mobile devices that would need to be worked on
Client Side	The cost would be similar to a Windows setup, as these operating systems are not open-source. The time required would depend on expertise—someone with experience	The cost of Linux would be minimal (if there is any cost at all), as it is open-source. However, it would require significant time and expertise, since Linux is not widely used. You	The cost would be similar to a Windows setup, as these operating systems are not open-source. The amount of time required would depend on expertise—someone with experience	Cost is unlikely to be a major concern with mobile devices. Experience may also be less of an issue, as mobile devices are generally easier to work with. However, more time would be

	using Mac would need less time, while someone with less experience would require more time.	would need someone skilled with Linux and allow them sufficient time to work, as it can be challenging even for experienced users.	using Windows would need less time, while someone with less experience would need more time.	required due to the variety of operating systems and devices that need to be managed.
Developing Tools	Swift is the most commonly used language for developing applications on Mac. There are several IDEs available for Swift, such as Atom, that can be utilized for development.	Eclipse and Atom are commonly used IDEs on Linux. Eclipse is primarily used for Java, although it can support other languages like C+.	Eclipse and Visual Studio are popular IDEs for Windows. Visual Studio can be used for development in languages such as HTML, C#, and JavaScript, among others.	The development tools are like those for Mac, and iOS apps are typically written in Swift. However, iOS and macOS differ in certain aspects.

Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**:
   For expanding Draw It or Lose It to other computing environments, Linux would be an excellent choice. It offers wide flexibility across various devices and platforms, is open-source, and provides cost-effective scalability. Linux supports many programming languages and frameworks, allowing Draw It or Lose It to be ported to different operating systems such as macOS and Windows with minimal modification.

2. **Operating Systems Architectures**: Linux operates on a monolithic kernel architecture, where the core system functions (e.g., process management, memory management, file systems, etc.) run

in a single address space. This architecture provides high performance and efficiency, which is beneficial for resource-intensive applications like Draw It or Lose It.

3. **Storage Management**: For storage management, Network-Attached Storage (NAS) or Cloud Storage solutions like Amazon S3 or Google Cloud Storage would be ideal. These systems offer scalability, remote access, and fault tolerance, which are crucial for a game that may involve large amounts of data and user-generated content.

4. **Memory Management**: On the recommended Linux operating platform, virtual memory management will be used to optimize how Draw It or Lose It allocates memory. Linux employs paging and segmentation techniques to manage physical memory efficiently. The operating system also supports memory overcommit to handle large workloads by temporarily using disk space when physical memory is exhausted. This ensures that the game can run smoothly even on lower-end devices without crashing due to memory limitations.

5. **Distributed Systems and Networks**: To enable communication between various platforms running Draw It or Lose It, a distributed system architecture using RESTful APIs or WebSockets would be effective. WebSockets will allow real-time communication between devices for multiplayer functionality. The network infrastructure must be designed to handle latency, outages, and connectivity issues by implementing techniques such as automatic retries, error handling, and caching mechanisms. A reliable and scalable cloud-based infrastructure (e.g., AWS, Azure) will ensure smooth interaction between devices, regardless of the platform being used. The system should also incorporate a load balancer to manage high traffic and prevent performance degradation during peak usage times.

6. **Security**: For securing user information on and between platforms, the Linux operating platform provides robust security features such as SELinux (Security-Enhanced Linux) for mandatory access controls and AppArmor for application-level security. Encryption should be implemented for data storage (using AES-256) and data transmission (via SSL/TLS protocols) to protect user data from unauthorized access.