Draw It or Lose It

CS 230 Project Software Design Template

Version 1.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	01/26/25	Edward McCauley	Module 3 submission

Instructions

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

Executive Summary

The Gaming Room currently has an application called "Draw It or Lost It" that is only available for Android. Their game is a modern version of the classic 1980's television game, "Win, Lose or Draw". Like the classic version of the game, their application will have teams of players competing against the clock and each other to solve the puzzles first. The game consists of four one-minute rounds where a team of players has thirty seconds to correctly guess a puzzle that is given in clues from rendered images from a stock library of drawings as the clock ticks down. If the team on the clock does not correctly solve the puzzle, the other teams have a 15-second clock to correctly solve the puzzle with the clues that were given.
The Gaming Room is looking to expand this Android application and develop a web-based version of the game. They are looking for Creative Technology Solutions (CTS) to streamline the software development process from Android to the web-based version. Before full implementation of the game will be approved by the client, an initial working version of the game application will be needed that correctly addresses software requirements of the game structure and play. After CTS managers have client approval of the software application, hardware specification requirements will be considered in the next phase of development.

Requirements

The Gaming Room is looking to expand on the success of the existing Android version of "Draw It or Lose It". The Gaming Room has not put an initial timeline on the release of the web-version of "Draw It or Lose It" because of their focus is quality products over quantity, this is why their games are so popular. They want CTS to focus on the fun and interactive aspects outlined in their requirements. They highlight that there should be only one game at a time. Each game can have one or more teams with multiple players playing during each instance of the game. The game and team names need to be unique to each instance of the game as well as a game clock for the challenge and steal rounds for teams will need to be implemented.

This game will serve as the test pilot for additional web-based versions of their other popular Android gaming applications. If the launch of the web-based version of this game is a success, they will consider moving forward with versions on additional operating platforms.

Design Constraints

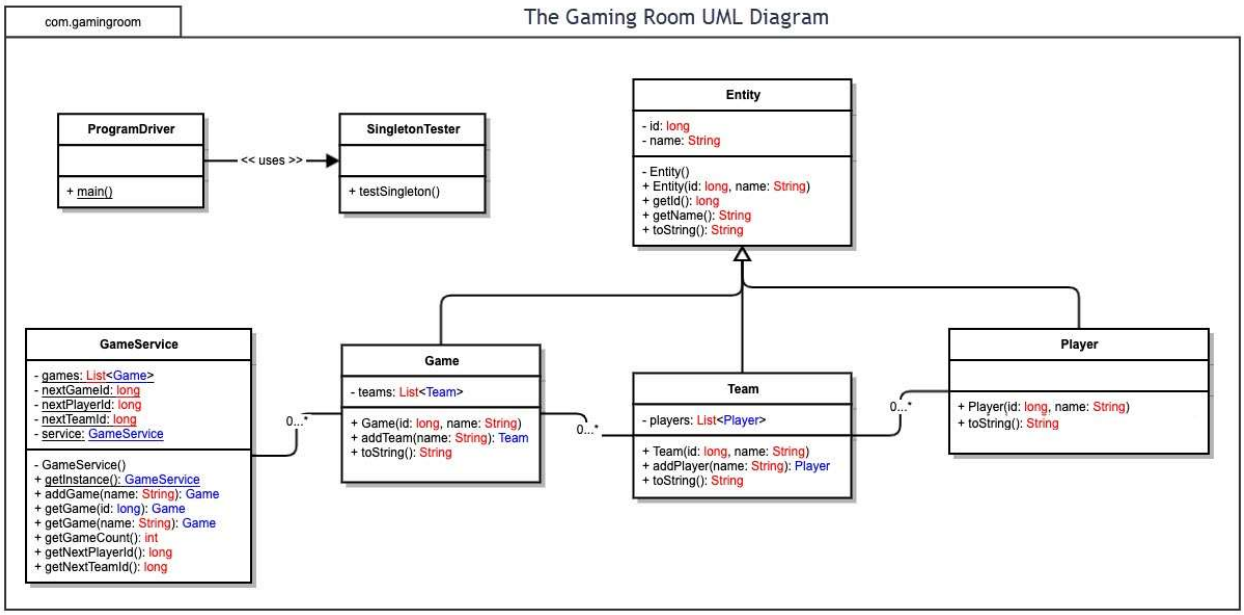
"Draw It or Lose It" is an interactive game. The clues to solving a single instance of the game by one to multiple teams with multiple players will need to run smoothly. The images in the library rendered as clues will need to be provided to the players without interruption. The premise of the game is to solve the puzzles with the pressure of the clock ticking away. If latency is present in the gameplay, this would allow for extra time for teams to potentially solve the puzzle and will detract from the game satisfaction. A balance between the quality of the image clues and the number of teams and players interacting may need to be considered in the initial software design and development before scaling multiple games instances. Additionally, this is a web-based version of the game therefore the advantages and disadvantages of programming languages and web browsers compatibility will need to be considered and proposed to CTS managers for client approval prior the next phases of development.

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 Gaming Room UML diagram represents the four main principles of Object-Oriented Programming (OOP) through encapsulation, inheritance, polymorphism, and abstraction. The Entity class is the base class for the Game, Team, and Player objects. It encapsulates the id and name attributes by restricting direct access to them but does allow for public methods to access them. The Game, Team, and Player classes inherit from the Entity class with additional functionalities to manage the game. Methods to create and add players, teams, and game names in addition to tracking unique names of each through id's. The Game and Team classes can create and handle multiple objects with Lists. The addGame, getGame, addTeam, addPlayer represent the abstraction principle of OOP. The GameService class creates and manages the singleton design pattern of the game. This class prevents direct instantiation of the game creation allowing for only one instance of a game and returns the singleton instance through the getInstance method. Additionally, it manages the singleton instance through the add and getGame methods as well as unique Player and Team methods. The ProgramDriver contains the main method that serves as the starting point of the game. The SingletonTester class tests for the singleton design pattern by ensuring that only instance of the GameService is created and used at a time.



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 has to 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 Devices
Server Side	Mac OS does not have a dedicated server operating system and is not typical in enterprise environments. Mac OS is developer friendly; however, resiliency and scalability are lacking compared to Linux and Windows servers.	Linux is the most common platform for large-scale web-based applications. Apache, Nginx, and Node.js use Linux OS which are capable of handling thousands of players. Linux OS is free to use and is open source with a large support community. They are also known for enhanced security measures.	Windows Server is more than capable of hosting large-scale web-based applications with the integration of ASP.NET. It is designed for enterprise-level applications making it both resilient and scalable. Microsoft regularly addresses security updates and patches. Licensing for Windows Server is costly.	Mobile iOS and Android would need to communicate with Linux or Windows servers to handle mobile application integration. Security is handled by OAuth and API's. Linux and Windows servers would handle the scalability and resiliency. There would be no licensing costs since mobile platforms are relying on these severs.

Client Side	Cost considerations for Mac OS clients are related to browser compatibility with Chrome and Safari. Front-end developers with expertise in Mac Sequoia/Monterey OS versions will be needed for the implementation and testing of platform and browser compatibility. Cloud and DevOps (Architects/Engineers) are required for optimization and scalability of Mac OS with the existing web platform. Security Engineers with expertise in Apple specific security protocols are needed if a Mac client option is chosen. The timeframe for Mac clients are comparable to the options with the focus on testing and security.	Cost considerations for Linux clients would include the expertise of Front-end developers testing Ubuntu, Fedora, and Debian and browser compatibility with Chrome and Firefox. The need for Cloud Architects and DevOps Engineers in Linux specific infrastructure, for example Docker and Apache or Nginx web servers will be used. Security Engineers with expertise in addressing specific Linux security vulnerabilities will be needed for the gaming application. As with Windows clients, a relatively short timeframe would be needed testing and fixes.	Cost considerations for Windows clients would include the expertise of Front-end developers testing Windows 10/11 OS and browser compatibility with Edge. The need for Cloud Architects and DevOps Engineers to ensure the scalability and resiliency of a thousand players. Additionally, the expertise of Security Engineers for Windows patches and updates. A relatively short timeframe is expected for Windows clients with an existing web platform in place that would be needed for testing, fixes and security updates.	Cost considerations for compatibility with mobile clients would be higher than Mac, Linux and Windows OS clients due to the performance optimization needs for the mobile UI/UX clients. There are additional costs for the Apple App and Google Play store for native app versions. Front-end and/or mobile designers and developers with expertise in both the mobile iOS and Android platforms would be needed to ensure a responsive web design for mobile specific features such as touch interactions, push notifications and on/off-line capabilities. Native app versions of the Draw It or Lose It would add more time and cost due to mobile features that need development and testing and the expertise of mobile engineers for scalability and security of the native app option.

Development Tools	HTML, CSS, JavaScript are the relevant programming languages for Mac clients. VSCode is a relevant IDE that supports React.js, Vue.js, and Angular framework choices for JavaScript libraries. Lighthouse for improving performance and quality of the web-based application. Safari Web Inspector and Chrome DevTools for testing the web platform on Mac OS specific browsers.	HTML, CSS, JavaScript are the relevant programming languages for Linux clients. Similar to Mac clients, VSCode is the relevant IDE choice, while React.js, Vue.js, and Angular are the common framework choices for JavaScript libraries. Chrome DevTools and Firefox Developer Tools offer testing and debugging tools across browsers.	HTML, CSS, JavaScript are the relevant programming languages for Windows clients. As with Mac and Linux clients, VSCode is the relevant IDE choice, while React.js, Vue.js, and Angular are the common framework choices for JavaScript libraries. Chrome and Microsoft's Edge DevTools assist with inspection and debugging.	HTML, CSS, JavaScript are the relevant programming languages for Mobile Device clients. In addition to React.js, Vue.js, and Angular frameworks, Bootstrap and Tailwind CSS frameworks offer a mobile friendly design option. VSCode is the relevant IDE option with Chrome DevTools and Safari Developer Tools assist with optimizing and testing of mobile device clients. If a native mobile app option is offered in the App/Play Stores, JavaScript and React are used for iOS and Android mobile devices. There are numerous options and tools available for hybrid or native app choices in addition to the web-based game application.

Recommendations

The following recommendations for the Draw It or Lose It gaming application are based on the best practices and industry standards that are consistent with similar highly popular and profitable massively multiplayer online games (MMOs) such as Minecraft, Fortnite, and Apex Legends. These recommendations are in line with The Gaming Room's requirements for expanding the current Android gaming applications to additional platforms, addressing latency constraints associated with the game as well as future scalability opportunities. Furthermore, rest assured that ensuring strong security throughout the entire software development lifecycle is a top priority at Creative Technology Solutions (CTS).

Operating Platform:

Cloud-based platforms providers AWS, Microsoft Azure, and Google Cloud are the top providers with the infrastructure to address scalability and latency in the gaming industry. The popular MMOs in the industry use these providers because of their proven track record handling the demands associated with high volumes of players and addressing latency concerns with their global data center deployment options.

Operating Systems Architectures:

Linux based operating systems such as Ubuntu and CentOS are popular options for similar Java based gaming applications. Top gaming companies such as Epic Games and Blizzard use Linux for their applications because of its stability, high speed performance and minimal overhead to handle large multi-user gaming applications.

Storage Management:

Major gaming MMOs companies manage their game assets and player data using cloud-based storage solutions like Amazon S3 and Google Cloud's storage. Cloud storage is a popular industry standard because it addresses the latency and scalability concerns associated with MMOs. Amazon DynamoDB is a database option that addresses the needs of a high performance and low latency gaming application, while Amazon RDS would handle the player and game management data.

Memory Management:

The recommendation to handle memory management for the Draw It or Lost It application is the Java Virtual Machine (JVM). JVM's automatic memory management allows for efficient resource utilization and optimization using garbage collection. Java based MMOs games such as Minecraft utilize JVM's memory management techniques to handle and optimize performance. Java's heap and stack management is the best approach and industry standard recommendation for The Gaming Room's application.

Distributed Systems and Networks:

AWS EC2 is a cloud service that offers multi-region server solutions that allows for low latency communication across mobile, PC, and console platforms. They automatically scale server loads based on demands and regional distribution. Fortnite and Apex Legends reduce the lag of their applications through distributed systems and multi-region networks through auto-scaling. Content Delivery Networks (CDN's) is a delivery system for players and gaming assets to ensure downloading is not interrupted.

Security:

Secure communication and data protection is the number one priority for gaming industry companies. AWS Shield is recommended to protect the infrastructure associated with web-based gaming applications. Games like Fortnite and Minecraft utilize TLS encryption for secure communication between clients and servers. OAuth 2.0 and Multi-Factor Authentication (MFA) are necessary implementations for user account and login management. DDoS protection, database encryption with regular backups and access control are all critical in protecting user's payment and personal data in any web-based application.