

**Draw It or Lose It**

# **CS 230 Project Software Design Template**

Version 3.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc159680139)

[Table of Contents 1](#_Toc159680140)

[Document Revision History 1](#_Toc159680141)

[Executive Summary 3](#_Toc159680142)

[Requirements 3](#_Toc159680143)

[Design Constraints 3](#_Toc159680144)

[System Architecture View 3](#_Toc159680145)

[Domain Model 3](#_Toc159680146)

[Evaluation 5](#_Toc159680147)

[Recommendations 8](#_Toc159680148)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 02/24/24 | Tremaine Rayner | Final draft for Gaming Room design. |

## [Executive Summary](#_sbfa50wo7nsh)

Our client The Gaming Room wants their game Draw It or Lose It to be developed on various platforms as a web-based game. Draw it or lose it is currently only available as an Android application. Developing this game on multiple platforms and marketplaces could be time consuming and costly during the development process. The CTS development team will adhere to guidelines for multiple marketplaces and platforms. Substantially with the code regarding the maintenance, power optimization, network, and compatibility will be required for this project to ensure equal user satisfaction amongst multiple platforms.

## Requirements

Draw it or lose it will be developed as a web-based game. Games and teams will have unique names with multiple players on each team. Unique names will allow players to see if a name is being used already. This game will have only a single instance running in the memory.

## [Design Constraints](#_2et92p0)

* Maintenance and updates on different devices. This will be a time constraint testing and deploying the game on different devices and operating systems. Connectivity and server maintenance on multiple platforms also must be maintained going forward.
* Draw It Or lose It must adhere to guidelines and policies of marketplaces like Apple and Android. Strict guidelines from a marketplace could cause a design constraint in development. The CTS development team will have to consider laws, data collection, and restrictions on content viewed by children during this process.
* Development for different screen sizes and resolutions. More time will be needed to develop for the abundance of different hardware specs like available in today’s market.
* The budget for development could be costly.

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

The UML diagram for the gaming room application shows different object-oriented principles such as abstraction, Inheritance, polymorphism, and encapsulation. Methods in the Gaming Room application can be used without detailing extensive information about this shows abstraction. Inheritance and polymorphism are shown in this UML diagram with the game, player and team classes inheriting from the entity class. With Entity being the parent class, any changes made within Entity will change the Game, Player, and Team classes. The ProgramDriver functions as the main method while the singleton class makes sure there is only one instance of the game. Encapsulation is also shown with the private constructors in the GameService and Entity class which are secure from other classes making modifications.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Tomcat and Apple Platform Deployment can be used to deploy this application on Mac.  Development for the application must be configured for Mac software and hardware capabilities.  Development for Mac can be costly and the Apple guidelines for their store are strict. Apple developer program costs $100 per year. | There are multiple deployment methods for Linux such as Kickstart and Ansible.  The are no licensing fees for developing Linux.  Linux is open source and not costly to develop for. | Windows Deployment services can be used as the deployment method.  Windows is susceptible to security threats. Following coding standards can prevent exploits being enabled through this application.  There are versions of Visual Studios that require payment.  Developer accounts for Windows will cost for a one-time registration fee. | Apple Platform Deployment can be used as a method to deploy the application on iOS.  Development for the application must be configured for iOS and android hardware requirements.  Company will have to pay for the Apple Developer program to deploy the application on iOS. Google Developer account will cost $25 to develop the application on Android. |
| **Client Side** | The application must be compatible with constant Mac software updates and hardware changes.  Google Chrome, Safari, Firefox, and Microsoft Edge are browsers supported on Mac.  Time constraint for developing multiple Mac devices.  Must work within the parameters of Apple guidelines.  Developers would need to be privy to Apple store guidelines. Local laws and age restrictions need to be considered. | Google Chrome, Firefox, chromium, and Brave are browsers supported on Linux.  Linux can be deployed on multiple hardware platforms. This could cause a time constraint when configuring the application with multiple platforms for Linux.  Linux is not as popular as Windows and Mac. The team will have to consider if it is worth developing for Linux while working with a budget. | Microsoft is constantly updating windows; the application needs to stay current with Windows updates.  Windows supports multiple web browsers including Microsoft Edge, Google Chrome, Firefox and many more.  Windows is a popular platform, developing this application for Windows could be profitable.  Windows development guidelines need to be considered during the development process. | The application must be compatible with different software and hardware changes for numerous mobile devices.  Mobile devices support multiple web browsers that include Safari, Google Chrome, Firefox, Opera, and Samsung Internet.  There could be a time constraint developing the app to meet screen and storage size requirements on different devices.  Different guidelines for multiple mobile devices need to be considered during the development process. |
| **Development Tools** | Development tools for mac includes Xcode, Apple developer program, Appium and other third-party tools.  Mac supports multiple programming languages such as Python, Java, C#, and Swift.  Some development tools are free to use, and some require a payment or subscription.  Multiple teams could be needed for this project. One development team can develop and communicate Mac hardware and software requirements to the entire team. | Development tools for Linux includes Manjaro, Ubuntu, Solus, and more for this open-source platform.  Linux supports multiple programming languages such as Java, Python, Golang, and PHP. These programing languages can be coded on multiple IDEs available for Linux.  Development tools are free for Linux and should not strain the budget.  Multiple teams would be needed for Linux because of the numerous platforms it uses. | Development tools for Windows includes Microsoft Visual Studio, Xcode, Windows App SDK, and Windows Community Toolkit.  Windows supports multiple programming languages such as Java, Python, C, Rust, and C++.  Some development tools are free to use, and some require a payment or subscription.  Multiple teams could be needed to develop for Window due to constant updates and coding difficulties developers could meet. | Development tools for mobile devices include Android Studio, Android Virtual Device, React Native, Xcode and other tools that support iOS and Android devices.  Mobile devices support multiple programming languages such as Python, Kotlin, Java, C#, and Swift.  Some development tools are free to use, and some require a payment or subscription.  Multiple teams will be needed for mobile device development. With a plethora of devices on the market the development team will have to configure the application to meet their hardware and security requirements. |

## Recommendations

1. **Operating Platform**: Draw It or Lose it application would benefit from Linux as the operating platform used for development. Linux is a well-known open-source operating platform that will provide great performance, security, and reliability that will allow for stability during the development cycle. With Linux being open source and free the the developer will have access to a plethora of tools and support while not straining the budget. Linux will also run on any hardware the development team will need for this application. The adaptability of Linux will allow for an easy transition if Draw it or Lose It is developed on consoles such as Xbox or PlayStation.
2. **Operating Systems Architectures**:

The operation architecture must include a client server model environment allowing for data to be stored and shared with users. I recommend that a compatible server be used with the Linux operating platform. Scalability of the Linux operating system with improvements through resources, tools, and community support will be impactful as this application grows. To host this server, I recommend using Hostgator which is a dedicated hosting server that is compatible with Linux. This server will be used to store media content, payment information, and user data that pertains to the game. The server side will verify user credentials upon a client-side request that would send encrypted data back to the client accessing the application.

1. **Storage Management**: For storage management, the draw it or lose it application can rely on hard disk drives (HDDs) or solid-state drives (SSDs), and a network attached storage to store data. SSDs are the block storage devices I would recommend. HDDs are cheaper compared to SSDs, but SSDs offer improved performance with faster access to files and low power consumption. In conjunction with SSDs, I recommend the use of a Network Attached Storage Device (NAS) for the sharing of data on a server. “Network-attached storage (NAS) is a file-dedicated storage device that makes data continuously available for employees to collaborate effectively over a network.” (What Is NAS Storage? - Network-Attached Storage Explained - AWS, n.d.). Draw it or lose it will be a challenge for the development team to handle multiple clients and properly communicate amongst themselves during the development cycle. The implementation of a NAS will only improve the way data is stored, accessed, and transferred between multiple users and devices.
2. **Memory Management**: The Linux operating platform is structured for memory management to enable the best performance possible. Virtual memory with Linux will allow the application to exceed ram capabilities. Memory that is not being utilized will be swapped to an SSD to free up space for the RAM. The pages that were allocated to the SSD can be sent back if needed with demand paging. If there are identical pages, then Kernel Samepage Merging will combine those pages into one creating additional space for memory. With RAM capabilities today I doubt there will be much need for virtual memory, but the team can rely on the technology to map memory usage. Memory protection is one more Linux memory management technique I wanted to mention. “Memory protection techniques prevent processes from accessing memory they are not authorized to access.” (Process Memory Management in Linux, n.d.). If unauthorized processes are accessing memory while authorized processes are running performance could be affected.
3. **Distributed Systems and Networks**: HostGator will be used to host this web application by storing data on the server side and distributing it to the client. HostGator offers a safe and secure server that provides high speeds, security through firewalls, and reliability. The Enterprise server, their most expensive option, offers 30GB of RAM and an unmetered bandwidth at $141.99 a month. If the cost of the Enterprise server is a concern, then other options will be explored. HostGator has an uptime of 99.9% which means this is a reliable hosting service and outages rarely occur. Users will need a stable internet connection to the server and play the game for the best experience. On the server side a stable connection will be needed as well to upload files and images to the server. The Shared hosting plan HostGator could hold data for the application on one server for clients sharing resources. For the Linux distribution I believe Ubuntu will be a great option. Draw it or Lose it does not seem to be graphically demanding so distribution through Ubuntu should not cause problems. Ubuntu is free to use, secure, and has development tools ready to use. The application can be deployed thorough the Ubuntu store where clients on different devices can access the game.
4. **Security**: The server through HostGator does offer protection with a firewall to prevent unauthorized access. This server is also protected from DDoS attacks. Data that is transmitted is also encrypted using the SSL certificate from HostGator. Encryption will provide security to user data such as passwords, credit card numbers, or game profiles for multiple clients. The Linux operating platform offers multiple features for security like secure boot, firewalls, intrusion detection but an antivirus will help improve security. For the Antivirus I recommend using Kaspersky to secure Linux. Kaspersky will provide protection from cyber security threats, that includes ransomware, malware, and phishing. This antivirus will also protect web traffic the detects malicious websites that could exploit any vulnerabilities. The developers will still have to adhere to coding standards on their end and not just rely on the security features from Linux. Secure coding, testing, and documentation will during the development cycle will prevent vulnerabilities for multiple clients.

**References**

What is NAS Storage? - Network-Attached Storage Explained - AWS. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/what-is/nas/>

Process Memory Management in Linux. (n.d.). <https://www.tutorialspoint.com/process-memory-management-in-linux#:~:text=Memory%20protection%20techniques%20prevent%20processes,permissions%20for%20each%20memory%20page>.

HostGator. (2020). Web Hosting. Hostgator.com. <https://www.hostgator.com/>

Server Protection Security Measures | HostGator Support. (n.d.). Www.hostgator.com. <https://www.hostgator.com/help/article/what-security-measures-are-used-to-protect-my-server>

Gillis, A. S., Peterson, S., & Lelii, S. (2021, July 12). virtual memory. Storage. <https://www.techtarget.com/searchstorage/definition/virtual-memory#:~:text=Virtual%20memory%20is%20a%20memory%20management%20technique%20where,technique%20used%20in%20a%20computer%27s%20operating%20system%20%28OS%29>**.**

8.4. Kernel Same-Page Merging (KSM) Red Hat Enterprise Linux 6 | Red Hat Customer Portal. (n.d.). Red Hat Customer Portal. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/virtualization_tuning_and_optimization_guide/chap-ksm#:~:text=Kernel%20same%2Dpage%20Merging%20(KSM)%2C%20used%20by%20the,systems%20by%20avoiding%20memory%20duplication>.

Ziani, S. (2021, October 19). Top 20 Reasons Why You Should Use Linux. Udemy Blog. <https://blog.udemy.com/why-use-linux/>

Endpoint Security for Linux | Kaspersky. (n.d.). <https://usa.kaspersky.com/small-to-medium-business-security/endpoint-linux>