

# Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/09/2021 | Andrej Oljaca | Updating the executive summary and design constraints |

**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](#_sbfa50wo7nsh)

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

The software design problem is that only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player. Another problem is that game and team names must be unique to allow users to check whether a name is in use when choosing a team name. To achieve this, I can check through the already existing game and team names and decide what to do based on what I find. Another problem is that each team will have multiple players assigned to it which can be achieved through a list. Finally, a game will have the ability to have one or more teams involved which can be achieved through a list.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

-Cost of cloud-based development environment to make it accessible through the web

-Use of specific programming languages to make it a web application

-The team I will have to build the app

The most common languages for a web based distributed system are HTML, CSS, and JavaScript for the front end and a variety of other languages including Java for the backend.

## [System Architecture View](#_ilbxbyevv6b6)

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](#_8h2ehzxfam4o)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

As seen in the UML diagram, the ProgramDriver class uses the SingletonTester class. The GameService class has a zero to many relationship with the Game class which is incorporated by the private List<Game> variable in the GameService class. The GameService class also has many private variables and many public methods, with the constructor being private, meaning GameService uses a Singleton pattern. There is only one instance of GameService at a time. These private and public methods and variables show a property of OOP called encapsulation. The GameService, Game, Team, and Player class inherit from the Entity class, which is another property of OOP. The Game class has a zero to many relationship with the Team class, and the Team class has a zero to many relationship with the Player class. Abstraction is shown through the use of methods which need to be called in order to do their function. Someone who does not understand code, does not need to know how the function works. Since there are many classes which have an “is a” relationship with another class, or are an “instance of”, these classes show polymorphism. Along with this, when the toString methods are overridden from the Entity class, this also shows polymorphism. As per the requirements only one instance of the game can exist in memory at any given time, and this is solved and shown in the UML diagram using an id in the Game class constructor. Also, the use of lists in some of the classes is how the other software requirements are met.

****

## [Evaluation](#_2o15spng8stw)

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** | <Evaluate Mac for its characteristics, advantages, and weaknesses for hosting a web-based software application.>  Mac OS X is built on Darwin – a Unix-like, open-source operating system developed by Apple and built on FreeBSD. This means that Mac users have access to free built-in server applications, like the Apache web server. With Apache and the DynDNS service, you can turn your Mac into a powerful web server. Apache is free. The amount you'll pay for a Mac OS X client/server would buy you a very capable Web server from Dell, HP, IBM, etc. Some cons of using Mac as a web server is that it tends to run slower than Linux or Windows according to my research. Also if you need to use Windows-specific technologies like ASP, .NET, MSSQL or Access on your website, it's probably easier to get a Windows web hosting package. | <Evaluate Linux for its characteristics, advantages, and weaknesses for hosting a web-based software application.>  Linux can also use Apache2 which is a free web server. Most web hosting companies prefer Linux because it helps keep the cost down, which makes it more affordable for their customers. A major benefit of hosting with Linux is the compatibility it has with other operating systems and software. You won’t experience any difficulties if you make a website with Windows and host it on a Linux web server. A Linux based server runs better than a Windows based server. Linux provides a flexible hosting environment with plenty of high performance applications. You can use both desktop and server applications, along with embedded systems. | <Evaluate Windows for its characteristics, advantages, and weaknesses for hosting a web-based software application.>  Microsoft costs money to be used as a web server. Windows Server runs services created and maintained by Microsoft. Compared to Windows Server, Linux handles more functions without issue, and doesn’t require reboots nearly as often. Microsoft has extensive documentation and live support. With Windows Server, you'll enjoy an easy-to-understand graphical user interface out of the box and Microsoft-backed customer support. Plus, Microsoft generally releases Windows Server drivers for new hardware in quick fashion. | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.>  Various server software apps are available for Android. However, many of these are out of date, intended for older versions of Android (such as PAW Server). Mobile servers are often implemented through open source software. Open source software is usually a good choice because it is free and maintained by a community of developers. A new iPhone app called "ServersMan@iPhone" turns the iPhone into a full-blown Web server. The name is said to be a tribute to Sony's "WalkMan." Once the app is installed, PCs on the internet can access the iPhone to upload or download files through a browser or they can use the webDAV protocol. |
| **Client Side** | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mac.>  Cost is a big issue when it comes to Mac clients. Macs are one of the most expensive PCs for their worth. Time is not a big issue because the Mac OS operating system is easy to pick up. Therefore expertise is also not a big hurdle when using Mac. It is important to use Cross-Browser Compatible Libraries and Frameworks. | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Linux.>  Cost is not a big issue for Linux because it is a free operating system, and the computers are usually cheaper then Macs. Time and expertise might be more of a problem however with Linux because it is not very beginner friendly due to all the terminal commands. It is important to use Cross-Browser Compatible Libraries and Frameworks. | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Windows.>  The biggest PC advantage is cost. Only the most high-powered and expensive PCs approach Apple products’ price point. Windows PCs have far more software created for them. Time is not a big issue and expertise is not as well because Windows is fairly user friendly. It is important to use Cross-Browser Compatible Libraries and Frameworks. | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mobile Devices.>  Cost is neutral for a phone based web server because phones are generally as expensive as computers. Time and expertise are not big issues however because there are many tutorials and articles on how to run a web server from a phone. |
| **Development Tools** | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mac.>  Most IDEs can be used across all operating systems. So Mac has access to most if not all of the most popular IDEs. For example Visual Studio Code and PyCharm. Mac can also support most programming languages. It has access to free web servers like Apache. | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Linux.>  Linux also has access to lots of important IDE’s such as Visual Studio Code and Eclipse. Also all the main languages run perfectley on linux. | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Windows.>  Windows is very compatable and hosts almost all of the most well known IDE’s. Along with this it works with most programming languages. This makes it highly attractive to developers. | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.>  A phone typically does not have IDE’s but uses apps and PWA’s to create a web server. You cant code on a phone. Along with this the two main phones Android and Apple differ in how they set up a server. |

## 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 The Gaming Room I recommend using Linux as an operating system that will allow The Gaming Room to expand Draw It Or Lose it to other computing environments. I selected Linux because it is cost-effective and can run on most computers and not just PC or Mac. Also, it is good as a server-side operating system due to its high performance. Linux can be downloaded from the web for free and installed on any computer.
2. **Operating Systems Architectures**: The Linux Operating System’s architecture primarily has these components: the Kernel, Hardware layer, System library, Shell, and System utility. The Kernel is the core of the Linux operating system and runs many major functions. System libraries are special functions that are used to implement the behavior and functionality of the operating system. The hardware layer consists of the CPU, RAM, and HDD. The Shell can be thought of as an interface. I recommend using the client-server architecture. I would choose this architecture because it is perfect for a software application that needs to run on multiple operating systems and clients. The server can remain the same and only the client needs to change.
3. **Storage Management**: There are a lot of filesystems available in Linux. The most commonly used ones are; xfs, ext4, ext3, ext2, btrfs and gfs2. So, files can be arranged on a disk in any of these following systems. For Draw It Or Lose It I would choose xfs since it is a good basic storage management system. It is a very flexible storage management system. It is also the default file management system. I would not create any partitions between Linux and another operating system, but rather use the whole space of the HDD to store the files and data. For the type of storage, I would recommend cloud storage due to its capability to expand easily as the user data grows. To start with the cloud storage will need at least 1.6 GB of storage for the images, and then will need to account for the user data.
4. **Memory Management**: Linux memory management is a complex system with many configurable settings. For Draw It Or Lose It, the data that is being accessed the most across all the games will be what is cached. These images that are cached will be a lot faster to access from RAM then from the HDD. In Linux everything that is going to CPU will go through RAM. Also to make things faster the CPU has level one, level two, level three cache. Any time the game accesses images it firstly needs to read the data from the disk and store them in RAM then depending on how often they are accessed. For Draw It Or Lose It I recommend around 32 GB of RAM since this is a good amount for most software.
5. **Distributed Systems and Networks**: Distributed Applications are software applications that can run on cloud computing platforms and that run on multiple systems simultaneously. They can also run on multiple different servers or computers called nodes. Usually, these distributed systems run on the same network. Distributed Applications are decentralized and less prone to attack. There needs to be good connectivity between the devices and outages need to be accounted for and handled appropriately. There needs to be a distributed outage management system. The client will communicate with the server using TCP/IP protocol suite. TCP is a connection-oriented protocol, which means a connection is established and maintained until the application programs at each end have finished exchanging messages. The client communicates with the server through something called requests and responses. The server receives a request, and the client receives the response. On top of that the client will communicate with the server through something called a REST API. The server will create routes and the client will be able to send POST, GET, and PUT requests as well as more to the servers API and receive responses.
6. **Security**: When coding the application, special care needs to be taken in regard to proper security. Authentication needs to built in in a smart fashion. Built-in libraries of the programming language, as well as external libraries, need to be used to ensure that proper authentication is achieved. Cookies and session timeouts need to be incorporated. Other attacks need to be stopped such as cross site scripting and phishing. From its very inception, security has been a cornerstone of the Linux operating system. Each user has to be walled off from others, and a password and user ID are required for an individual to use Linux.

Sources

<https://www.macinstruct.com/tutorials/how-to-turn-your-mac-into-a-web-server/>

<https://itxdesign.com/why-choose-hosting-with-a-linux-web-server/>

<https://www.makeuseof.com/tag/turn-android-device-web-server/>

<https://technewscollector.com/turn-your-iphone-into-a-web-server.html>

<https://cybernews.com/best-web-hosting/windows-hosting/>

<https://www.crucial.com/articles/mac-users/mac-vs-pc-the-pros-and-cons>

<https://www.elprocus.com/linux-operating-system/>

<https://tekneed.com/storage-management-in-linux-explained-with-examples/>

<https://www.golinuxcloud.com/tutorial-linux-memory-management-overview/>

<https://www.investopedia.com/terms/d/distributed-applications-apps.asp#:~:text=Distributed%20Applications%20(%C4%90Apps)%20are%20software,run%20on%20multiple%20systems%20simultaneously>.

<https://searchnetworking.techtarget.com/definition/client-server#:~:text=Clients%20typically%20communicate%20with%20servers,end%20have%20finished%20exchanging%20messages>.