

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 3.0

## 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)4

[**Evaluation**](#_2o15spng8stw)5

[**Recommendations**](#_m8aleynsvzvc)12

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/10/2022 | John Brungard | Completed the Executive Summary, Design Constraints and Domain Model sections. |
| 2.0 | 07/27/2022 | John Brungard | Completed the Evaluation section. |
| 3.0 | 08/04/2022 | John Brungard | Completed the Recommendation section. |

**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)

The Gaming Room wants to develop a web-based game called Draw It or Lose It that serves multiple platforms. Software requirements of the web-based game include having one or more teams involved, having multiple players per team, having unique names for games and teams, and having only one instance of the game to exist in memory. However, Draw It or Lose It is only available as an Android app currently. Creative Technology Solution proposes creating an emulator for Draw It or Lose It, which will allow the game to be played on Android and Non-Android Devices whilst upholding the requirements given by the Client.

Emulation presents many advantages such as being an inexpensive solution in domains such as maintenance, being a Fast Data Transfer solution, which allows developers to test and debug programs quickly and effectively, and allows adaptability such as being able to use different device controllers for the game as well as being able to run other games concurrently with Draw It or Lose It. However, emulation has its drawbacks too, even in a web-based environment, which may concern performance and speed. These concerns are addressed in the Design Constraints and Evaluation sections of this template.

An emulator is suggested in the context of this project due to its proficiency to test how software interacts with a combination of hardware and software. It will help detect whether firmware updates will cause problems and will also help see the performance of the program using different central processing units and memory allocations. Furthermore, because Draw It or Lost It depends upon visual quality being that it will be using rendered images, emulation will help deliver better FPS and resolution compared to simulation while being more cost effective than using real devices.

## [Design Constraints](#_2et92p0)

1. Emulation is known to have performance issues due to requiring larger amounts of processing power to function smoothly. Therefore, designing the web-based game will require attention to detail concerning memory usage such as preventing memory leaks within the program.
2. Emulation involves writing code for an entire CPU in Software, which can lead to increased design and programming sprints and in turn, a longer deliverable date. This is necessary for the emulator to be fully functional in different web-based environments.
3. Emulators will not always reflect the specific hardware and software features of each supported device. Therefore, designing and programming must make use of universal methods and other practices so that the emulator can be upgraded and expanded as updates on supported devices arise. An example of this is the Singleton Pattern where the instance Service will be the only instance of the game to exist in memory by creating unique ids for each instance of game, team and player using iterator patterns.
4. A game must have the ability to have one or more teams involved.
5. Each team must be able to have multiple players assigned to it.
6. Game and team names must be unique.
7. Only one instance of the game can exist in memory at any given time.

## [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)

The ProgramDriver class is the driver of the other classes in the program. Specifically, it is used to run functions and test for the Singleton Pattern within the program.

The Singleton Class tests a singleton’s behavior. Specifically, it is testing whether only one instance was created within the program.

The Entity Class holds common attributes and behaviors universal to the Game, Team and Player Classes and acts as a parent class as to obtain its methods.

The Player Class inherits from the Entity Class and is responsible for holding information about a player to include their id and name.

The Team Class also inherits from the Entity Class and is responsible for holding information about a team to include their id and name. The Team Class is also responsible with managing all of the players in Draw It or Lose It through its addPlayer() method and players arrayList.

The Game Class is the final class that inherits from the Entity Class and is responsible for holding information about a game to include its id and name. The Game Class is also responsible with managing all of the teams in Draw It or Lose It through its addTeam() method and teams arrayList.

The GameService Class is responsible for setting up the Singleton Pattern using the Class Object Service. It holds all the ids for each component of the program and increments them when appropriate. It also allows only one instance of the game to exist in memory at a given time, which is checked or added through its methods.

The GameService, Game, Team, and Player use a zero to many associations so they may have 0 to an unlimited amount of instances when checking unique names, ids and adding games, teams, and players.

ProgramDriver has a usage dependency on SingletonTester for output purposes within the program, displaying the id and name of each game, team, and player added into the program.

**"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)

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** | Advantages  -Apple offers a discrete graphics card that uses its own high-speed memory called Video Random Access Memory (VRAM). This frees up the main CPU and puts less of a strain for RAM, so high-quality games can be run sufficiently.  -Macs include an unlimited OS X Server option for a relatively low one-time cost ($999 US as of Summer 2022). This option is ideal for expanding companies who project having more than 10 clients.  -Mac’s “Apple Maintenance Program” allows purchasers of the unlimited OS X Server to receive updated versions of it following 36 months from purchase for the cost at which you bought the server. This not only allows the Gaming Room to stay up to date but gives the option to install copies of each version of the server.  Disadvantages  -Many of Apple’s processors include an integrated Graphics Processing Unit (GPU). This makes it harder to run high quality video games due to the GPU sharing the RAM meant for the Central Processing Unit (CPU).  -Because Apples Maintenance Program involving the purchase of an OS X Server only guarantees free newer versions of the Server for up to 36 months, you may end up getting multiple versions of the Server or end up with 1 additional version of the Server depending on the frequency at which an updated version is released.  -Mac only offers one other option for a Server purchase besides the Unlimited Server option. It is called the 10-client license and only allows up to 10 clients. This is not ideal for the Gaming Room and leaves only one ideal choice for a Server purchase. | Advantages  -Linux offers open-source operating systems, meaning its source code is available for any programmer to improve. While open source may seem like a security threat, it has proven to identify and eliminate security threats more so than its competition.  -Linux restricts root access using the technique of least privilege. Only superusers can use all the privileges, which is also advantageous against attacks. This contrasts with most other systems which give any user admin privileges.  -Linux offers server licenses through Red Hat Enterprise, which has a multitude of options depending on your needs with many options being free. For the Gaming Room, notable options would be The Developer Suite, Workstation, Developer Workstation, and Server.  Disadvantages  -Server Distribution costs with Red Hat Enterprise are based on yearly subscriptions, which can be less cost friendly in the long run depending on your chosen option.  -Licensing costs with Red Hat Enterprise can be expensive with a Standard Server annual subscription costing $799 US and a Premium one being $1,299 US. While these costs fare better against some competition, it also does not account for desired available add-ons such as extended update support ($249) and high availability ($399). | Advantages  -Windows has fixated its focus on building upon and improving Windows 10 with free updates. One of these updates is including a feature that allows it work with Linux applications, allowing programmers to use a wider variety of applications for their developing purposes.  -Windows Server options come with many advantages for ease of use for the server side. This includes managing server storage collectively instead of drive-by-drive, a backup feature configured for server files, and integration and implementation of hosted services and apps.  Disadvantages  -Microsoft offers server distribution and licenses through Windows Server. Out of the competition, this tends to be the most expensive option. Viable options for the Gaming Room range from $1069-$6155 US.  -Although the WindowsOS is very popular, this comes at a disadvantage of usually being the target of attacks and breaches. To the credit of Windows, this is not always due to the OS being less secure than their competition, but rather the positive correlation between number of users and number of attackers. | Advantages  -Cloud Servers offer a high degree of IT support which can help in situations such as reducing the downtime during outages.  -Mobile devices support mobile emulation, which are virtual devices that can mimic real devices and user actions. This typically makes them totally compatible with current and older systems.  Disadvantages  -While Cloud Servers excel in mobile access, their online status makes them more susceptible to breaches and attacks on data security and privacy.  -As previously stated in this document, a primary concern of using emulation resides in its speed to reproduce behavior. This may cause problems on a client side, especially if coupled with mobile networks further affecting the behavior of the game. |
| **Client Side** | Advantages  -Macs offer a Parallels Software that allows Mac users to run Window applications with MacOS. This work-a-round to Mac not having ample services gives users an increased valued experience to play Draw It or Lose It along with other games not available before using Mac’s powerful technologies.  -Apple uses a cloud computing service to distribute functions over multiple data centers, which allows a viable option for compatibility amongst web browser platforms.  Disadvantage  -Mac does not share the plethora of games and services available compared to its competition such as Windows. This may lean users towards using another operating system for gaming.  -Cloud Computing generally has a high degree of difficulty to implement, which may require the help of senior developers or trained cloud computing individuals.  -Cloud Computing also can be seen as an expensive option with the average Cloud Server being $15,000 US monthly (Spring 2022). | Advantages  -Linux open-source OSs have not only been an advantage in security but allows programmers and users to be highly customizable and flexible, which can lead to increased value and less constraints that would overall reduce total time involved in a task.  -Linux offers many options pertaining to versions of their operating system. These are called distributions. Distributions offer a variety of options that compromise between resource usage, added features, and graphical elements.  Disadvantages  -Distributions of Linux are often known to be more complicated to learn compared to other operating systems such as Windows. This means it will take time to learn how to navigate the OS and this time is based on your experience with Linux and technology. You may also need multiple teams with one being IT support.  -While there are a vast amount of distributions Linux offers, which is typically a positive thing, it also presents problems in choosing which distribution best suits your needs, experience, and budget. This can increase the cost, time and experience needed to create cross platform game depending on the solution. Fortunately, you can change distributions quite easily by installing the new one in the same space. | Advantages  -Windows is compatible with most low-resource computers, which welcomes more customers constrained to a lower budget.  -Windows is famously known for its ease-of-use operating systems, which decrease the time and expertise needed to perform tasks.  Disadvantages  -Many programs that come with WindowsOS are paid such as graphics software and download manager. If the current or future games are developed from the Windows Operating System, you may require a high budget with the summative costs of all components necessary for a finalized game.  -Some options of Windows Server do not include Active Directory Federation Services. This means users will not have the option to access gaming applications with a single sign-on | Advantages  - Emulation presents many advantages such as being an inexpensive solution in domains such as maintenance, being a Fast Data Transfer solution, which allows developers to test and debug programs quickly and effectively, and allows adaptability such as being able to use different device controllers for the game as well as being able to run other games concurrently with Draw It or Lose It.  Disadvantages  - Emulation involves writing code for an entire CPU in Software, which can lead to increased design and programming sprints and in turn, a longer deliverable date. This is necessary for the emulator to be fully functional in different web-based environments.  -Emulators will not always reflect the specific hardware and software features of each supported device. Therefore, designing and programming must make use of universal methods and other practices so that the emulator can be upgraded and expanded as updates on supported devices arise. An example of this is the Singleton Pattern where the instance Service will be the only instance of the game to exist in memory by creating unique ids for each instance of game, team and player using iterator patterns.  -Cloud Computing generally has a high degree of difficulty to implement, which may require the help of senior developers or trained cloud computing individuals.  -Cloud Computing also can be seen as an expensive option with the average Cloud Server being $15,000 US monthly (Spring 2022). |
| **Development Tools** | -MacOS uses Xcode as a language  interpreter for C++, Python and Java. It also includes Apple exclusive languages such as Swift and Objective-C.  -Xcode and many other IDEs exclusive to Apple are usually free for all MacOS users, although a yearly subscription of $99 US is needed to distribute an app amongst the different app stores.  -Alternative MacOS exclusive IDEs include AppCode and CLion. The option for IDEs specifically geared towards MacOS along with the versatility of using other popular IDEs like Visual Studios makes for a versatile and easier approach to creating cross-platform applications.  -Swift is based on C languages (most popular languages for creating games) and is focused on Apple devices. It is user friendly and is open source, making it ideal for mobile developers, especially when paired with other IDEs like Atom to expand to cross platforms. | -Most Linux distributions come preinstalled with Python and is available as a package on all others. Pythons’ extensive library is useful when designing games that require adaptability and additional features. It is also one of the easier languages to comprehend and implement.  -Linux offers IDEs and code editors that are specifically dedicated to one language. For instance, IntelliJ IDEA is used to write and edit Java code while IDLE is focused on Python programming.  -Many IDEs and Code Editors come preinstalled with a Linux depending on the distribution, which also proves cost effective. Other IDEs presented for Linux are cross platform IDEs also available for other operating systems such as Mac and Windows. | -Windows offers an IDE in the form of Microsoft Visual Studios. It offers many editors pertaining to most computer languages such as HTML, CSS, JavaScript, JSON. Java programming languages are particularly effective for implementing Object Oriented Programming.  -There are multiple versions of Visual Studio that can be purchased based on the needs of the purchaser. Visual Studio Enterprise, meant for larger systems, has a monthly subscription of about $250 US.  -Though Visual Studios is the most popular IDE pertaining to Windows, there are many other options such as Code::Blocks and Eclipse. The quality and versatility of most of these IDEs lowers the chance of a technical constraint. | -IOS devices typically run similar languages to Mac such as Objective-C, Swift, C#, HTML5 and Java.  -Androids offer many major programming language options too in Python, C/C++, and Java.  -There are a variety of Android Development IDEs that support core languages in mobile app development such as HTML, Java, and JavaScript. Most can run on a variety of devices including Windows, MacOS and Linux. Some examples include NetBeans, Komodo, Cordova and PhoneGap.  - There is an official IDE for Android called Android Studio that was built by Google. This IDE attempts to tackle technical constraints with code templates, an intelligent code editor, and an emulator for frequent testing. However, it still contains unresolved issues in system lag, high system requirements and heavy RAM usage.  -The official IDE for IOS devices is called XCode made by Apple. It is equipped with a UI creator and allows ease of use for profiling and heap analysis. However, technical constraints lie in its ability to only work with Apple OS, no support for tabbed work environments and little information online to solve problems due to a Non-Disclosure Agreement set forth by Apple. These constraints may deem necessary a hired expert on XCloud and any problems it may present. |

## 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**: Based on the requirements of Draw It or Lose It along with the constraints and domain associated with it, I would recommend using Windows 10 as the base operating system for development.
2. **Operating Systems Architectures**: Windows 10 offers the 32 and 64-bit architectures. The 32-bit architecture is more adaptable with older hardware while the 64-bit generally offers high performance, security, and features. In terms of OS architecture, Windows uses two modes, user, and kernel, as shown in the following:

Graphical user interface, application

Description automatically generated

(Credit: Grm wnr & Xyzzy n)

Kernel mode gives access to the hardware and executive resources. It is within the executive section that memory is managed, system calls are implemented, and I/O devices are configured. It also deals with different power stages of the device such as off and on. Furthermore, the Kernel is responsible for bootstrapping and is called upon by the process manager for the initialization.

User mode contains system processes and Dynamic Link Libraries. It contains the Win32 subsystem which is responsible for running the 32-bit applications (Win64 would run 64-bit applications). The OS/2 subsystem runs OS/2 applications but is not present in newer systems as it was removed in Windows 2000. POSIX is now replaced by the Windows Subsystem for Linux.

My recommendation is to use Win64 due to increased performance, more available Random Access Memory and offering more security.

1. **Storage Management**: There are two appropriate recommendations of storage. One being Solid State Drive (SSD) and the other being Hard Disk Drive (HDD). SSD, known as a common replacement to HDD, uses solid-state flash memory through a flash controller and memory chips. They do not use physical spinning parts. HDD on the other hand is a spinning disk with a read and write magnetic head. It uses spinning parts for read/write options.

I am making the recommendation to use SSD. Being that it is a replacement of HDD, it has been

recorded to have faster read and write speeds, requires less power to handle tasks, offers

quicker startup and comes with more types for specific storage needs. These include the basic

type (described briefly above), non-volatile memory express (higher transfer speeds for high

demand environments), and Peripheral Component Interconnect Express (higher performance

than the basic option but may need a custom driver).

1. **Memory Management**: Windows 10 employs many memory management techniques so that data and information can be stored and used correctly.

One such technique is called memory compression. This technique is specifically made to compress pages of memory after the memory manager confirms that a certain percentage of the available RAM for the system is being used. When a page becomes compressed, it is then transferred to the compression store where it is not contributing as much to the overload of memory usage. Then when a process needs to use that page again, it is decompressed and utilized. This is very efficient concerning the performance of a device and will allow memory taken up from a page to be used for another process when not in use. This will help manage the multiple processes to go into a game like Draw It or Lose It.

Windows also employs memory protection in where a process is protected by its own virtual address space. This can include having some address spaces allowing a read-only option. This is combined with Copy-on-Write Protection where more than one process maps their virtual address spaces to the same physical memory pages until one of them modifies the page. This makes it so processes are not activated until necessary.

1. **Distributed Systems and Networks**: The distributed System would be the internet and would be coupled with a Cloud Service or Virtualization that is connected by virtual networks. Both have their advantages and disadvantages when deciding to host a web-based game. Given the circumstances of Draw It or Lose It, I would recommend going with a Virtualization approach over Emulation or a Cloud service.

Virtualization has virtual machines that are used instead of servers and devices. Windows 10 has a feature to create virtual machines in which a host system runs multiple guest operating systems which will help solve The Gaming Room’s desire to host a game on different platforms. Virtualization also offers save states and a flexibility of available test environments, which allows developers to test and debug programs quickly and effectively, which will prove very useful as the game progresses and uncertainties arise. Finally, Virtualization is a very cost-friendly option compared to a physical system. This means money that would have been used for this purpose can be put towards other features, solutions, or potential add-ons.

In terms of unexpected events such as an outage, Virtualization proves to recover much quicker than a traditional system. When disasters strike a traditional system, an IT admin or tech must replace or fix it which can take at the very least a few hours. However, Virtualization makes it to where you can clone the Virtual Machine that was affected, which can take a few minutes when handled correctly. It also provides a process called backup virtualization where the host can create a restore on a virtual machine.

1. **Security**: A security measure I believe will be crucial in protecting user information between platforms is Role-Based Access Control. Role-Based Access Control, also known as Role-Based Security is where different roles are established for different operations. The authorization to utilize an operation is based on your role. Take a google document for example. There are different roles such as document owner, editors, or viewers. There are different roles assigned to each user and what functions they are authorized to use. An editor can change the text in the document, but a viewer cannot.

A complemental security feature to Role-Based Security would also be the technique of least privilege. When assigning privileges to a role, it is usually best to give that role “just enough” privileges to perform their duty or operations. Giving any more privileges beyond the necessities involved with their role would cause instability and a higher chance an attack would leave more damages.

Furthermore, a two-step verification method would be ideal in protecting the user’s information. As the name suggests, the user must provide two authentication methods to access their data and information. The methods can vary such as providing a password and then receiving a code on a cell phone or email address which must be copied onto the website for access.

If two-step verification is coupled with requirements to set up a strong password or code for log in, it makes for an extremely viable security option. A strong password or code would be at least 8 characters in length, contain at least one uppercase letter, lowercase letter, digit, and special character, and should not be a word in the dictionary. These specifics are all to help against a breach, particularly through enumeration.