

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 3.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/16/2023 | Kat Gilliland | Added executive summary, design constraints, and description of domain model |
| 2.0 | 07/30/2023 | Kat Gilliland | Shared an evaluation of the different operating systems: Mac, Linux, Windows, and mobile devices |
| 3.0 | 08/13/2023 | Kat Gilliland | Shared operating platform, storage management, memory management, distributed system, networking, and security recommendations for continued development and launch of Draw It or Lose It |

**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)

Draw It or Lose It, a game offered by The Gaming Room, is currently only available for Android devices. In this game, teams compete against each other to guess what is being “drawn” by the application. The game allows for multiple players and teams and consists of four one-minute rounds. During each round, the application renders a drawing that is fully complete at the thirty-second mark. The team up against the clock must guess what the image is before the time is up. If they do not answer the puzzle correctly, the remaining teams have a fifteen-second opportunity to offer one guess each. The Gaming Room would like to create a web-based version of this game so it can be used and played on various platforms.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

* *Game must be able to run on various platforms. This means the game must be designed to work on multiple platforms and operating systems, each with its own unique set of limitations.*
* *Game must assign unique identifiers to multiple teams, players, and games. This means the game must be able to check and validate whether or not the team name, player, or game already exists in memory as only one instance of the game can exist at any given time.*
* *Each team must have multiple players. The game should check and validate whether there are enough players on a team before starting up.*

The design constraints above must be considered when developing a web-based version of Draw It or Lose It. It is vital to The Gaming Room that the game is designed to be accessed and played on virtually any platform or operating system; because of this, it is important that the game is developed using a programming language that can operate on various systems.

## [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 UML class diagram provided below demonstrates many of the object-oriented programming principles. For example, inheritance is shown by the Game, Team, and Player classes. In the diagram, these three classes inherit common properties and behaviors (such as id and name) directly from the Entity base class. Encapsulation is shown through the use of private methods and attributes, which are denoted by the – and + symbols. The Game service class shows a direct 0 to many (0..\*) relationship with the Game, Team, and Player classes. This means that the game can have 0 to multiple games, teams, and players. However, it is important to note that there will be a unique identifier attached to each of these items as only one instance of each can exist at any given time. The ProgramDriver class runs the main and uses the SingletonTester class. The SingletonTester class checks and validates the game’s unique ids to ensure that only one instance of each exists.

**"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** | Mac is known to be user-friendly and has a simple and easy-to-use graphical user interface (GUI). Mac is also known to be a very secure operating system, boasting less malware and security issues than other operating platforms, such as Windows. | Linux has more processing power than the other operating systems, therefore, it can handle more functions and tasks at a faster rate. Linux is also extremely flexible and can be configured to meet specific server needs. The sky is the limit with customization. | Like Linux, Windows is highly customizable and compatible with more hardware compared to other operating systems such as Mac. Similar to the Mac operating system, Windows also has a very straightforward and clean graphical user interface (GUI), however, it can be argued that Windows can be more complicated to use due to the way it stores its programs and apps in folders. | Mobile devices are known to be faster and more convenient to use as they can be taken on the go. Mobile devices are also more accessible since they allow web apps to be accessed by any user regardless of their choice of operating system. Since web apps on mobile devices are not native to any single operating system, they may not be able to access special features unique to a particular OS. |
| **Client Side** | Mac is more costly than other options because it is a propriety operating system created by Apple and will come with licensing fees. Mac’s costs are similar to Windows.  There can be a bit of a learning curve for those accustomed to using other operating systems, but realistically should be simple to pick up and implement. | Linux is the most affordable option because it is an open-source operating system. It requires a good amount of technical expertise to use and will take a bit of time and work to learn all the ins and outs, however, once there is a solid understanding of Linux, it is simple to maintain and requires less upkeep. | Windows is more costly than other options because it is a propriety operating system created by Microsoft and will come with licensing fees. Windows’ costs are similar to Mac, however, Mac is known to be a bit more expensive. Very basic technical expertise is required to use the Windows operating system. | Web applications for mobile devices are slightly more difficult to implement than web applications for a particular operating system and therefore, may be more costly and more time-consuming to build and maintain. |
| **Development Tools** | Many different programming languages can be used to deploy the Gaming Room’s software on Mac. Most commonly, developers use Objective-C and Switch languages with this operating system, however, C++, Python, and Ruby can also be used. The IDEs most used with Mac operating systems are Xcode, CLion, and Pycharm. | Linux supports many languages natively; however, it most commonly supports languages such as C/C++, Java, Python, JavaScript, and Swift. IDEs used with this OS are Eclipse, NetBeans, IntelliJ IDEA, Visual Studio, and Pycharm. | Windows primarily supports the following languages: C, C++, and C#. As far as IDEs go, Windows can run Visual Studio, NetBeans, JetBrains Rider, and Android Studio. | If the Gaming Room decides to deploy the software to different mobile devices, it will be important to use a cross-platform programming language. Cross-platform programming languages include: JavaScript, Java, React Native, and Kotlin. The following IDEs can be explored for these languages: Eclipse and IntelliJ IDEA. |

## Recommendations

1. **Operating Platform**:

To expand their game, Draw It or Lose It, to other computing environments, my recommendation to The Gaming Room would be to use the Windows operating platform. Because of Windows’ straightforward graphical user interface (GUI), it is known for being extremely user friendly and will be easier for The Gaming Room to navigate when managing Draw It or Lose It or working on new projects. In addition to ease of use, Windows is compatible with most hardware and software; an aspect of the OS that will come in handy should The Gaming Room needs to work with specific hardware or software to expand upon or make modifications to the game.

1. **Operating Systems Architectures**:

The Windows operating system is based on a layered architecture and is divided into two different modes of operation: kernel mode and user mode. Kernel mode is where the OS kernel and other low-level processes operate. Kernel mode is a privileged processing state in which a process or thread can access all system resources and is able to perform any task. The kernel has complete control of the entire system. User mode, or application mode, is a restricted processing state in which a process or thread is not able to access certain system resources or perform certain tasks. To access kernel mode resources or to perform specific privileged operations, user-mode applications need to make system calls.

1. **Storage Management**:

Windows offers a useful feature to manage drive space called Storage Sense. Storage Sense automatically frees up drive space by removing things like unnecessary or temporary files and items in the recycle bin. Storage sense is simple to use; it can be toggled on or off at any time and can even be configured to run whenever the user deems fit. Storage sense can be set to either run only when the system is low on space or can be set to run periodically (every day, week, month, etc.).

1. **Memory Management**:

Windows uses a memory management technique known as virtual memory that will be very beneficial to the Draw It or Lose It game. Virtual memory is a storage allocation scheme that is implemented using both system hardware and software. Virtual space is limited by the size of a system’s virtual addresses; to increase memory capacity, virtual memory permits the operating platform to map between these virtual addresses and physical addresses in the RAM or disk. Virtual memory is a particularly useful memory management approach for Draw It or Lose It, a game with multiple users and many graphics, as it will allow the program to use more memory than is physically available and enables the execution of multiple programs simultaneously. Another advantage of utilizing virtual memory is that it helps prevent crashes and improves system stability by allowing the OS to handle memory shortages.

1. **Distributed Systems and Networks**:

To effectively enable the game, Draw It or Lose It, to communicate between various platforms, I’d recommend that The Gaming Room continue its development using the software, Unity. Unity is a real-time development engine designed to deliver the necessary environment and tools for creating multi-platform, multiplayer games. Unity IDE is renowned for its intuitive features and quick learning curve. Offering many online tutorials and documentation, Unity walks game developers through every step of the process. Through Unity, a game can be exported to 17 different platforms, including the commonly used iOS, Android, Windows, and Linux. Unity does not currently offer its own networking solutions; however, it is known to integrate well with a third-party networking service known as Forge Networking. Forge Networking is a free, open-source game networking system designed specifically for cross-platform, multi-user games. By utilizing a 3rd party networking system, The Gaming Room can focus its efforts on the game itself as Forge Networking would take responsibility for any issues that may arise, such as connectivity issues or outages.

1. **Security**:

The Gaming Room can protect its users' information on and between various platforms by enforcing the following security measures: two-factor authentication and data encryption. Two-factor authentication will require the player to provide two distinct forms of identification to prove they are who they say they are and that they should be allowed access to the account; this adds another layer of protection to the information they have stored inside the game. Encryption protects the confidentiality of digital data by translating it into a secret code that can only be unlocked or accessed using a unique digital key. Encrypting data prevents it from being accessed by malicious actors or affected by cybersecurity threats.