

Draw It Or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## 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  1.1  1.2 | 05/18/23  06/24/23  06/17/2023 | July Wellman  July Wellman  July Wellman | Initial design proposal  Development requirements  Recommendations |

**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 is an Android app game based on Win, Lose, or Draw, where teams compete to guess what is being drawn. A game consists of four rounds of play lasting one minute each. Drawings are rendered at a steady rate and are fully complete at the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit. The staff at The Gaming Room would like to develop a web-based game that would serve on multiple platforms based off of Draw It or Lose it but do not know how to set up the environment. The game will have many players with one or more teams playing in the same game instance. Due to this required mechanic, the use of a singleton pattern in the software will be required to ensure that the objects will only have one instance. An iterator pattern will also be used to traverse the data structures aimed at enhancing performance.

## Requirements

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* 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.

## [Design Constraints](#_2et92p0)

* Only one instance of the game can exist at any given time.
* The game must be web-based and available on many platforms.
* Names must be unique for each player in the game instance.

A singleton pattern will be needed due to the mechanic that only one instance of the game can exist at any given time. Draw It or Lose It is currently only available through the Android app, so we may look into a REST API to help with the transition and communication between the different platforms.

## [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 diagram below shows the design of the main classes in the game Draw It or Lose It. The game is developed following principles of object-oriented principles. The entity class acts as the superclass and includes member functions inherited from the game, team, and player classes. The game service references the game class, then the game class references the team class, and then references the player class. These references show an association from one class to another. The SingletonTester class is for the singleton pattern design and the program driver has the main() method, therefore is the driver of 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** | -Flexible terminal commands to configure server, access, or make changes.  -Every user has the ability for web hosting and has options for different web hosting requirements.  -There is a licensing cost associated with Mac. | -An open-sourced cost-effective OS.  -No licensing fees to host.  -Linux is able to host web applications, but cannot host at the same scale as Apple or Windows. | -More software compared to other OS.  -Windows is the dominant platform so it naturally has more features and options for various web hosting.  -Windows has poor tech support and it is more susceptible to viruses. | -More popular and has high portability.  -Portability adds to a much wider reach.  -Not as strong as its other competitors due to hardware.  -There are licensing fees for both Android and IOS operating systems. |
| **Client Side** | -Mac has many programs that only work for their systems requiring team members to all be on the same platform.  -The cost of the initial setup can be expensive, and there may be monthly recurring charges. | -Linux is the most secure next to Apple.  -Linux is the most difficult to set up due to the lack of programs available.  -Open-sourced format makes Linux the most cost-effective option. | -Has the highest compatibility rate with other applications making Windows the preferred option.  -Cost is similar to Mac and forced updates can take time.  -Not as secure as Linux and Mac. | -Very cost-effective and user-friendly for the user.  -Both Apple and Android use their own language so development makes take a longer time.  -Can be costly to maintain applications. |
| **Development Tools** | -Swift  -Notepad++  -HTML  -CSS  -Javascript  -Python  -Ruby  -Java | -Visual Studio  -Eclipse  -Notepad++  -HTML  -CSS  -Javascript  -Java  -Python  -C++ | -Visual Studio  -Eclipse  -Notepad++  -HTML  -CSS -Javascript  -Java  -Python  -C++ | -Visual Studio  -Eclipse  -Notepad++  -HTML  -CSS  -Javascript  -Java  -Python  -C++ |

## 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**: Microsoft Windows is the recommended primary platform for Draw It or Lose It. Windows offers a wider range of features than other platforms, and is compatible with a wide range of hardware and software solutions, making Windows the ideal choice for Draw It or Lose It. Windows also has an easy-to-use UI that makes it simple for novice and experienced staff to adjust to learning.
2. **Operating Systems Architectures**: Windows operating platform architecture is a complex system designed to provide a secure, reliable, and efficient computing environment. The architecture is designed to ensure that all components of the system interact properly and efficiently. Windows provides an easy-to-use interface for users to access their data and applications. It also offers various security features such as authentication and encryption to protect user data from malicious attacks. The architecture includes the kernel, memory management, device drivers, user interface, and many more.
3. **Storage Management**: Local storage on the hard drive can be used and cloud storage is available as well for mobile storage between devices. Cloud storage would allow the client to not have to maintain in-house servers that would require expensive upkeep.
4. **Memory Management**: The creation of the game will require a database or library of pictures. The memory allocation allows for easy storage of pictures. This will also allow for the client to be able to keep the whole project together in a secure area on the computer.
5. **Distributed Systems and Networks**: IDEs like Unity and Unreal engine are popular choices for gaming applications. These two IDEs enable cross-platform game creation and can be run on the Windows operating platform.
6. **Security**: Windows comes with built-in security protection software. The system will scan for malware, viruses, and security threats in real-time and updates the system automatically to keep the system and the client/user safe. Other programs can also be purchased to provide added protection to the system.