

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 | Apr 14, 2025 | Andres Torres | Project 3 |

## [Executive Summary](#_sbfa50wo7nsh)

CTS was tasked by The Gaming Room to design a web-based version of their Android game, Draw It or Lose It. The game should support multiple teams, each with multiple players. Each game instance, team, or player should be unique.

To achieve this, a singleton creation pattern has been adopted to prevent the creation of multiple game instances. Additionally, an iterator pattern has been implemented to ensure that conflicting teams and team members are avoided.

## [Design Constraints](#_2et92p0)

The Gaming Room already has an Android-based app of the Draw It or Lose It game. CTS has been tasked with extending this to the web, so the game needs to be compatible with web deployment. Java has been chosen to simplify the web deployment.

Any existing APIs that run the Android app need to be reviewed and updated for mobile usage.

## [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 application’s core component is the main class, which initiates the creation of games, teams, and players. The process is handled by the GameService class, following the singleton design pattern to ensure only one instance of GameService exists at any given time.

Once GameService is ran, the driver class can call on the addGame() method. This method uses the iterator to avoid the creation of duplicate Game objects. The newly created Game object is then added to the List games.

Similarly, the addTeam() method, used to add teams to the game, and the addPlayer() method, used to add players to the team. They also utilize the iterator pattern to prevent the creation of duplicate Team and Player objects. These newly created objects are then added to the respective Lists, teams, and players.

The designed UML diagram shows various object-oriented programming techniques. Polymorphism and inheritance are used in the extension of the Entity class and the constructors.

**"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 OS X provides more flexible commands in the terminal to configure the server and determine access permissions. However, these commands are limited in hardware options exclusively provided by Apple and can be more expensive. | The most reliable and stable option is also highly cost-effective because it is available with a wide range of hardware options. | This software is compatible with a wide range of hardware and release options, and it also supports HTML files. | It would be challenging to both code and store information for an extended period. Therefore, alternative options would be more suitable. |
| **Client Side** | Apple’s products are priced at a premium, needing the end user to make a purchase. The software’s versatility is limited unless a virtual machine is employed to load it into a different operating system. | Although Linux is cost-effective, as most users are not familiar the operating system, it would require a great amount of time to learn and master. Nonetheless, it is available on a wide range of devices. | Most users are familiar with Windows, and the software is user-friendly. Additionally, it is available on a wide range of devices. | Constrained by its design and battery duration, it provides clients and users with timely updates. |
| **Development Tools** | Objective-C is the primary programming language utilized on macOS. | Supports a wide range of programming languages, including C, C++, CSS, Java, JavaScript, HTML, PHP, Perl, Python, Ruby, and Vala. However, it does not support all programming languages. | This platform supports a wide range of programming languages, including C, C++, CSS, Java, JavaScript, HTML, PHP, Perl, Python, Ruby, and Vala. However, it does not support all languages. | Swift is widely employed for iOS development, whereas Java is the preferred language for Android devices. |

## 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**: Given that the application is already available on Android devices, I recommend releasing it on Windows as well. This could be accomplished by listing the application on the Windows app store. Since the application is written in Java, it is well-suited for working with and writing on Windows computers.
2. **Operating Systems Architectures**: I recommend a personal computer equipped with an X86 AMD processor, which possesses the capability to simultaneously host the application’s development and execution processes without crashing.
3. **Storage Management**: For running the game on an SSD, the cost has become significantly more affordable. However, when developing the application, it is crucial to have multiple copies and backups.
4. **Memory Management**: Windows is the preferred operating system for both development and execution. The amount memory available for Windows enables all threads of a process to access its own virtual address space.
5. **Distributed Systems and Networks**: For multiplayer games, especially those that use a network, it is common for these games to share a database among users. This serves as a valuable tool in mitigating duplicate matches and optimizing server resources. Additionally, I would favor programming the game in Java to ensure its compatibility with a wide range of operating systems.
6. **Security**: Encapsulation serves as a protective measure for sensitive cases. It protects information within the program. In accordance with the General Data Protection Regulation (GDPR), programs are mandated to minimize the data collected during the processing and utilization of personal information.