

# Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 11/07/2023 | Stanislav Kovalchuk | Modifications were implemented on the cover page, document revision history, executive summary, design constraints, system architecture view, domain model, and recommendations. |
| 1.1 | 11/21/2023 | Stanislav Kovalchuk | The characteristics, advantages and weaknesses of each of them were evaluated for presentation to the client. |
| 1.2 | 12/04/2023 | Stanislav Kovalchuk | The characteristics and methods typical for various system architectures were analyzed and recommendations were given to The Gaming Room. |

**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 aims to create a cross-platform web game titled "Draw It or Lose It," inspired by the 1980s television game Win, Lose or Draw. Currently available only for Android, the game features four rounds, each lasting a minute. Drawings are rendered steadily and are fully completed by the 30-second mark. If a team fails to guess the puzzle within the allotted time, other teams get a chance to provide one guess each within a 15-second limit. The goal is to expand the game's availability to various operating systems.

## Requirements

The client wants this game to be cross-platform, which means that the user can play it on different devices.

## [Design Constraints](#_2et92p0)

The software introduces specific design constraints, including the requirement to implement a singleton pattern to ensure only one instance can exist at a time. It supports the inclusion of multiple teams and players within predefined limits. Additionally, the software is designed to run on different operating systems, although its current version is limited to Android devices.

## [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 Entity establishes a connection between the Game, Team, and Player classes, implying that they inherit or retrieve information from Entity. This relationship can be depicted in UML through inheritance, making Entity the superclass. Examining their associations, Team and Player exhibit a "has a" relationship. Specifically, Game has a Team, and GameService has Games. In UML notation, this is denoted as aggregation (HAS-A). When we say "has a," it signifies an instance of one class having a reference to an instance of another class. In this diagram, GameService holds a reference to Games, Games holds references to Teams, and Team holds references to Players.

**"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 offers effortless accessibility and server configurability, boasting an intuitive graphical user interface that simplifies tasks. Additionally, it provides flexibility through its adaptable terminal commands, catering to users with varying levels of expertise. | Budget-friendly. Platform navigation can be challenging. It features a command shell for straightforward server configuration and enhanced accessibility. | The server side comes with a hefty price tag. It offers a user-friendly graphical interface and includes a command prompt for advanced users. | Other devices outperform in terms of specifications. Mobile device specifications differ from one user to another. |
| **Client Side** | Costly for users, demanding moderate time and expertise. Accurate skills are essential to navigate the operating system effectively. | Significant expertise and time investment are necessary. Proficiency in Linux data is required to operate the system. Linux users face a substantial cost. | Pricier than Linux systems, but user-friendly with straightforward support for Windows setup. Requires minimal expertise to learn and operate. | Offer clients and developers the flexibility to access updates from any location. Implementation might be slightly more challenging compared to other devices. |
| **Development Tools** | Languages such as HTML, CSS, and JavaScript are utilized. Libraries are available to aid in front-end development. Additionally, there are other development tools like PyCharm, GitHub, and Visual Studio, among others. | Languages include HTML, CSS, and JavaScript, supported by various libraries for front-end development. In Linux systems, additional languages like JavaScript, Ruby, PHP, and Python are also available. | Programming languages such as HTML, CSS, and JavaScript form the foundation. Various libraries provide support for frontend and other languages. Developer tools like Eclipse, command prompt, and PyCharm are commonly used in this context. | Programming languages like HTML, CSS, and JavaScript are fundamental components. Libraries are available to bolster both front-end development and other languages. Integrated Development Environments (IDEs) for programming include HTML, PHP, C++, and Python. |

## 

## 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**: To enhance the game application, I suggest considering the Windows OS as the operating platform. Windows enjoys the highest usage among the three platforms, with 45.3% of the population using it. Given the web-based nature of the application, Windows would be an optimal choice for handling backend tasks such as user requests and troubleshooting. While license costs need to be taken into account, the platform offers greater flexibility in the long run.
2. **Operating Systems Architectures**: Windows OS features a user-friendly graphical interface, designed for effortless navigation. Moreover, it supports a diverse array of Integrated Development Environments (IDEs) for application development. Popular IDEs compatible with Windows include Visual Studio, Eclipse, IntelliJ, JetBeans, and several others.
3. **Storage Management**: Windows OS simplifies storage management with easy-to-configure settings, enabling seamless navigation for backend operations. While owning a Windows cloud server entails an initial expensive investment, it proves worthwhile as the space can be expanded in the future. This investment is particularly valuable if the client plans to expand the game application, add diverse assets, and accommodate future developments.
4. **Memory Management**: Windows OS offers two memory options: physical and virtual. Virtual memory, the more preferable choice, efficiently handles large programs. Unlike physical memory, virtual memory provides essential features like memory protection and extended usage for physical memory, making it advantageous for various applications.
5. **Distributed Systems and Networks**: When operating within the Windows OS environment, it's likely to encounter some common issues such as lagging, recurring queuing problems, and overloaded servers that can impede basic tasks. I believe these challenges can be addressed effectively through open communication and collaboration among the developer team members.
6. **Security**: Numerous safety features are in place, including the continuous operation of the Windows Defender antivirus program on the client's computer. VPN services are also included to enhance user information protection. It's crucial to conduct daily security checks to safeguard users and prevent unauthorized access to their information. Additionally, developers working on the application should be well-educated in securing and encrypting user data to ensure comprehensive protection.