

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_heading=h.gjdgxs)1

[**Table of Contents**](#_heading=h.30j0zll)2

[**Document Revision History**](#_heading=h.3znysh7)2

[**Executive Summary**](#_heading=h.2et92p0)3

[**Design Constraints**](#_heading=h.tyjcwt)3

[**System Architecture View**](#_heading=h.3dy6vkm)3

[**Domain Model**](#_heading=h.1t3h5sf)3

[**Evaluation**](#_heading=h.2s8eyo1)3

[**Recommendations**](#_heading=h.3rdcrjn)5

## [Document Revision History](#_heading=h.3znysh7)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 11/13/22 | Lawton Miller | Executive summary, design constraints, project summary and other design 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](#_heading=h.2et92p0)

This application is an online hosted multiplayer drawing and guessing game. Game format is based on “Win, lose, or draw”. Players and games will have unique ID’s assigned to them to ensure an organized system. This game will have to be hosted on servers to enable our multiplayer gameplay.

## [Design Constraints](#_heading=h.tyjcwt)

Each game must have unique instances on the platform, as well as unique identifiers for players and teams. Ideally this game will be able to be played cross platform on more than one operating system.

## [System Architecture View](#_heading=h.3dy6vkm)

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](#_heading=h.1t3h5sf)

The UML Diagram provides a visual for the different classes relate to each other. One thing that will stick out here is how The Game, Team, and player classes all inherit from the Entity class. The program driver which contains the main structure calls the singleton tester which ensures that only one instance of each game can exist in the system's memory at a time. The GameService class uses public methods to retrieve and set attributes from other classes such as team, game, and player ID’s

**"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. n**

## 

## 

## [Evaluation](#_heading=h.2s8eyo1)

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** | Not as popular as a server. There is not a dedicated mac os for server functionality, but has flexible terminal commands for configuration. Expensive | Very Popular Server solution, many open-source versions that can be used for various applications. Flexibility is a major plus. | Popular server choice, guaranteed support, and better usability for those familiar with windows. Resource intensive with use of detailed GUI | Not a very viable Server option, typically lacks computing power to be an effective server. |
| **Client Side** | Very popular consumer machine. Games and applications can be played on browsers such as chrome and safari as well as downloaded from the App store.  More expensive | Not a common user interface, would take certain expertise to use and maintain program  Minimum cost | Very Common consumer OS. By developing with this OS in mind you are covering your bases for many consumers  Medium cost | Mobile gaming is incredibly popular so developing an app in this space is important. More difficult to implement. There are two major platforms to consider, iOS and Android, so both versions would have to be developed. |
| **Development Tools** | One of the many strengths of Macs. Many applications on Mac OS run Swift, an open source language developed for the platform. Macs can run any language such as HTML/CSS/JavaScript, and would likely save some time on other platforms to do so.  Eclipse, visual studio available + many more | Capable of running most popular languages such as  HTML/CSS/JavaScript. IDE’s such as Eclipse and visual studio work well | Capable of running most popular languages such as  HTML/CSS/JavaScript. IDE’s such as Eclipse and visual studio work well but with a more user friendly GUI than LINUX | Many frameworks have been built to ensure fit and compatibility on many types of 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**: I think that this game should be hosted in a windows environment due to ease of use and the tools available for development and maintenance.
2. **Operating Systems Architectures**: Windows provides an extremely user friendly GUI that abstracts computer functions in a very familiar way. When playing or hosting games this allows a positive user experience. Applications developed and ran on windows off an ease of use in development by allowing a vast array of dev tools as well as easy testing if you are developing in the same environment that the game will be played
3. **Storage Management**: Windows machines can be capable of file storage on a large level, and are easily managed with the file explorer in windows. Local networking is made easy with windows, and files can be easily shared throughout an organization with modifiable privacy settings that can allow permissions for some users and workstations but not for others.
4. **Memory Management**: This program will contain a large library for the various images that the game will use. Windows OS is able to dynamically move data to and from your memory from storage reliably so that your game can run smoothly.
5. **Distributed Systems and Networks**:

The requirements to run games cross-platform on each OS will require the ability for players to access and manipulate data that is communicated to their device from the server. The server must then process and communicate these changes to the other users nearly instantly. By having very similar software on each platform the installed game should be able to run this game on the server efficiently.

1. **Security**: Windows comes with built in security to protect data, however when running an enterprising business that is responsible for users data, you will likely want to use another third party tool as well as having dedicated personnel responsible for security.