

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.2 | 04/14/22 | Justin Lombardi | Updated Recommendation section |

**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 wants to develop a web-based game, Draw It or Lose It, that serves multiple platforms. Currently, it is only available in an Android App. Gameplay includes four rounds lasting one minute each. Drawings are rendered and fully complete at 30 seconds. If team doesn’t guess the puzzle before time is up, the remaining teams have the chance to offer one guess each to solve the puzzle with a 15 second limit.

## [Design Constraints](#_2et92p0)

* A game will be able to have one or more teams
* Each team will be able to have multiple players assigned to them
* 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.

The constraints listed above are for gameplay constraints. In other aspects of the design, this must also be able to run on all devices. This includes multiple operating systems as well including Apple, Linux, and Windows. We will need to either be able to re-write the code for Apple devices or run the code on other devices by inheriting other languages

## [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 class creates a bond between the Game, Player, and Team class. Those three classes inherit and extend information from the Entity class. In the UML, the empty arrowheads that link the Game, Player, and Team classes to the Entity class shows Inheritance. We can also see the zero to many relationships between the GameService, Game, Team, and Player as well. One game can have many teams and one team can have many players. It also shows that only one GameService can exist in memory at a time.

**"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 the server, access, or make changes. Mac OS X server is available and inexpensive but not as popular as Windows or Linux for these specific tasks. | Like the Mac, but more cost friendly and is also open source which would make for additional resources. | More software available compared to other OS. They also offer a Windows server which might be more expensive, but it would work. | It's better if the server is immobile and can be tracked at a single place. Specs are better in other devices. Also, phones do not have the power to handle a full-on server compared to a computer. |
| **Client Side** | Moderate expertise and time required. Cost can be comparable to Windows, possibly a bit more. | Maximum expertise and time required but can require minimum cost. Linux is not commonly used like Mac and Windows so there would be more of a learning curve or have someone coming in with previous experience. | Minimum expertise and time required with minimum cost out of Linux and Mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | Macs can run a lot of languages including HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. | Linux can work with VS Code and Eclipse (possibly Atom). Languages can include HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. | Very similar to Linux. VS Code and Eclipse to name a few of the many IDE’s. Languages include HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. | You can create countless apps using React Native or even Swift. Both languages and software can be run on all three machines. |

## 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 would recommend using Windows operating platform because it is less expensive and can be used with minimum expertise. Since the App is already available on Android, it should pair nicely.
2. **Operating Systems Architectures**: Windows can provide services used by all windows-based applications that enable applications to show a Graphical User Interface (GUI), while accessing system resources. These services can be used using a user account or a server.
3. **Storage Management**: Windows now comes with a nice feature called **storage sense**. This allows you to manage files on your hard drive, along with how much space it takes up. Other features include being able to choose save locations for apps making it easier to find. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost or carelessly deleted.
4. **Memory Management**: The memory allocation allows for easy storage of pictures outside of the default picture folder. This allows you to keep your whole project together in a more secure area on your computer. This includes when you’re working with your IDE and opening files from it to create the game.
5. **Distributed Systems and Networks**: Unreal Engine is an example of a Cross-Platform Game Development Software that is incredibly powerful. To meet the needs of creating this for multiple devices, this is what I would recommend. To prevent other problems like outages or connectivity, the company will need to make sure their servers are strong enough to support large player volumes along with backup power for power outages.
6. **Security**: Windows comes with a built-in security protection software. Though to secure user data and information it would be recommended to use another source. Though if we are talking about what is on the machine windows comes pre equipped with protection. This system scan for malware (malicious software), viruses, and security threats. This all happens in real time, and because threats change the system updates automatically to keep the system and user information safe.