

Win, Lose or Draw

# **CS 230 Project Software Design Template**

Version 1.3

## 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 | <01/20/23> | Idriss Kacou | Brief description of changes in this revision |
| 1.2 | <02/01/23> | Idriss Kacou | Development Requirements Update |
| 1.3 | <02/12/23> | Idriss Kacou | 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)

The Gaming Room intends to develop a web-based game, "Draw It or Lose It", that can be played on multiple platforms. Currently, the game is only accessible on android devices. The objective of the game is for multiple teams, each consisting of several players, to compete in four rounds, each lasting one minute. A picture is randomly chosen from a library, and one team member tries to guess the image before time expires. If the image is not correctly guessed, each member of the opposing team gets an opportunity to answer within 15 seconds.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

The game requires the participation of at least one team, with each team comprising multiple players. Unique game and team names must be assigned, allowing users to check for availability.

Only one instance of the game can be active at any given time.

The game must be designed to function across multiple platforms.

The development of "Draw It or Lose It" game must include the following specifications:

The game must involve one or more teams

Each team must consist of multiple players

Unique game and team names are required for user identification and availability check

Only one instance of the game can exist at any time

The game must be compatible with multiple platforms including android, Windows, Linux, and Apple devices.

In addition to the game aspect, the Gaming Room also aims to make the application available on all devices. This includes adapting the code to work on other mobile devices and computer operating systems, potentially through the use of multiple programming languages to create stronger and more versatile code.

## [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 creation of an Entity class establishes a relationship between the Game, Team, and Player classes through inheritance. This means that the classes share common attributes such as "name" and "id" as Entity acts as a superclass. Using UML diagrams, we can depict this relationship as Team and Player being a "has a" type relationship, while Game has a Team and GameService has Games. This is referred to as aggregation in UML diagrams. A user "has a" reference to an instance of one class and a reference to an instance of another class. As depicted in the UML diagram, the GameService holds a reference to Games, Games holds a reference to Team, and Team holds a reference to Player.

**"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 allow for server configuration, access, and changes. It is widely used in web hosting but has drawbacks, making it less favored for web hosting services. Despite this, it offers upgradeability and various options for diverse web hosting needs. | Linux offers the same benefits and is even more cost-effective. Known for its security and popularity, it is the preferred choice for web hosting services. However, it may be challenging to find applications that support the necessary web hosting requirements. | Compared to other operating systems, there is a wider availability of software. Dominant in the market and closely integrated, it offers high resource requirements, quick loading times, and high user comfort. However, its downside includes a high susceptibility to viruses and limited technical support. | A server that is stationary and can be monitored in one location is preferable. Although other devices may have better specifications, this option is more popular and highly portable. Its benefits include wider accessibility, improved compatibility, and cost-effectiveness. However, it may be limited to only a few mobile devices and lacks strong security. |
| **Client Side** | Moderate levels of expertise and time investment are necessary. The cost is comparable to Windows. | The highest level of expertise and time investment is required, while the cost is kept to a minimum. | Low levels of expertise and time investment are needed. The cost is comparable to that of a Mac. | Offers flexibility to clients and developers to access updates from anywhere. However, implementation is slightly more challenging compared to other devices. |
| **Development Tools** | Macs can run Swift, a popular choice, along with helpful tools like Notepad++. All programming languages, including but not limited to HTML/CSS/JavaScript, can be executed on Macs, as well as libraries that support frontend development and general-purpose languages like Java, Python, PHP, and Ruby. | Linux supports visual studio, eclipse, and Notepad++ for a user-friendly experience. It also offers numerous additional languages and tools. Programming languages supported include, but are not limited to, HTML/CSS/JavaScript and libraries for frontend development and general-purpose languages like Java, Python, PHP, and Ruby. | Windows is easier to use than Linux but can run the same languages, including visual studio, eclipse, and many others. Notepad++ is a user-friendly tool among the multiple options available. Programming languages supported include, but are not limited to, HTML/CSS/JavaScript, and libraries for frontend development and general-purpose languages like Java, Python, PHP, and Ruby. | Countless apps can be developed using both Android and Swift, both of which can be run on all three types of machines. Programming languages supported include, but are not limited to, HTML/CSS/JavaScript and libraries for frontend development and general-purpose languages like Java, Python, PHP, and Ruby. |

## 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 suggest that The Gaming Room begin on Windows devices, as it offers a wider range of software and requires less expertise and cost to get projects off the ground. Additionally, you will not face a lack of Integrated Development Environments to work with.
2. **Operating Systems Architectures**: The Windows operating system offers various services that support the functioning of applications that run on the platform. These services include the ability to display a graphical user interface, access system resources, and provide support for graphics and multimedia, messaging, and web services. These services can be utilized by both individual user accounts and servers.
3. **Storage Management**: Windows 10 offers a convenient tool called Storage Sense, which enables you to monitor and organize the files on your hard drive, including their storage usage. It also provides the option to set custom saving locations for apps, making them easier to locate. Additionally, you can utilize the cloud for data storage. The integrated storage management system simplifies the process of creating and storing large projects, ensuring they are protected from loss or accidental deletion.
4. **Memory Management**: In order to develop this game effectively, you will need to build a database or library containing a large number of images. The memory allocation feature enables convenient storage of these images outside of the default picture directory, providing a secure location for your entire project, including when using the IDE and accessing files for game creation.
5. **Distributed Systems and Networks** As every operating system differs from each other, I explored ways to release the game that would run on all platforms. I discovered Develop 4, which facilitates cross-platform game development. It's a versatile Integrated Development Environment that can be used on any device. Once the game is developed, you can easily export the game file to the web, iOS, Android, and many other platforms that support cross-play, resolving dependencies. To avoid issues like downtime or connectivity problems, the company must ensure that their servers have sufficient strength to handle high player traffic and have backup power in case of power outages.
6. **Security**: Windows comes equipped with built-in security protection software. While this is useful in securing user data and information, it is still recommended to use an additional source for maximum protection. Nonetheless, Windows has its own pre-installed security measures, such as real-time scanning for malware, viruses, and other potential threats. The system automatically updates itself to address new and evolving security threats, ensuring the safety of the user's information and system.