

Win, Lose or Draw

# **CS 230 Project Software Design Template**

Version 1.0

## 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 |
| --- | --- | --- | --- |
| 1.0 | 11/14/2021 | Ido Tal | Created executive summary, Design Constraints and Domain model |
| 2.0 | 11/28/2021 | Ido Tal | Filled in the evaluation for Mac, Linux, Windows, and Mobile devices |
| 3.0 | 12/12/2021 | Ido Tal | Filled in the 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 company wants to create a web-based version of the gaming app: Win, Lose or Draw. There are a set of rules that the game is required to have as part of the client’s request. The game must render images from a library of stock images and it must consist of four rounds play while each round lasts for one minute. Drawings must be rendered at a steady rate and completed by 30 seconds and if the team fails to guess the puzzle before the time expires then the remaining teams have a chance to guess the puzzle but within a 15 second time limit. The Gaming Room company does not have a staff that is capable of setting up the proper game environment for the game, and for that reason it seeks help.

## [Design Constraints](#_2et92p0)

* The game will have the ability to have one or more teams involved. This will be done through a public add method to allow for the potential of adding more teams.
* Each team will have multiple players assigned to it. This will be done through a public add method to allow for the potential of adding more players.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name. This will also be accomplished by using the iterator pattern to make sure that there are no teams and players with the same identifiers.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player. Additionally, the use of the singleton creation pattern will make sure that only one instance of the game can exist.
* The game needs to be web-based. Java programming language could be used to create the game and it can be tested across multiple web browsing apps to check that it works.

## [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)

This application demonstrates the use of encapsulation and inheritance through the following. The Entity base class contains properties and methods that are inherited by the Game, Team, and Player classes. For the application to run, each class needs a unique name and id. The Entity class has the public methods: getId() and getName() and that gives access to those private properties to the Game, Team, and Player classes. The GameService class has zero to many associations with the Game class. The Game class has zero to many associations with the Team class, and this allows the game to have one or more teams involved. The Team class has zero to many associations with the Player class, which allows for each team to have multiple players. The iterator design makes sure that every team name and game name is unique, while the singleton creation pattern means that only one instance of the game can exist at any one 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** | **Advantages:**  - Cheap, it costs $19.99.  - User friendly graphical user interface.  - It is easy to access and configure the server.  **Disadvantages:**  - Can only support Apple’s OS, any hardware use for the server side has to be Mac. | **Advantages:**  - Free and open source.  - Large variety of specific OS to choose from.  - Can be used on almost any device  **Disadvantages:**  - Harder to use as it does not always have a graphic user interface making it less intuitive. | **Advantages:**  - Windows has a large variety of options with different features based on needs and costs.  - Windows features a graphic user interface that makes it easier to work with.  - Can support running both Linux and Windows side by side.  **Disadvantages:**  - Windows is more expensive, and the cost can vary based on needs and features. | **Advantages:**  - A server on a mobile device is portable.  - There are Linux distributions that support using a mobile device as a server, giving some variety to a the person hosting setting up the server.  **Disadvantages:**  - Hardware limitation makes this option undesirable, as it will not have the capacity to handle a large volume of traffic. |
| **Client Side** | To test on a Mac OS than a Mac OS is necessary, and since they cost more than many Windows computes of similar hardware then this might drive cost up. Development needs to be done for Chrome, Safari, and Firefox. To Develop a web based app it is important to know HTML, CSS, and JavaScript. | Linux is free and that will drive cost down, and it can be installed on any computer which means it can be test on any device, but Linux will require someone with experience and knowledge to properly test on a Linux device. Apps must be developed for Firefox and Chromium as they are the more popular web browsers on Linux. Must know HTML, CSS, and JavaScript. | Apps must be developed for Microsoft Edge, Firefox, and Chrome. The large variety of free IDEs means that it will not cost a lot of money, but the developer will need to know HTML, CSS, and JavaScript. | Testing on a mobile devices requires an android device and an iOS device which will increase the cost. The two devices use different IDEs and each device will have to be tested and this will also increase the time. A developer who is familiar with both devices is needed and that can use HTML, CSS, and JavaScript. Additionally, the time to develop might increase since the app web based app must take into consideration differences in screen sizes and changes in the mobile devices orientation. |
| **Development Tools** | To develop for Mac there is a need to use Xcode which is Apple’s integrated development environment. Xcode does support Swift, C, C++, Java, Python, Ruby, and more. | Linux has a large amount of IDEs and tools for developing a variety of software. Eclipse could be used for developing apps using Java. Geany could be used for Python. Visual Studio code is good for HTML, CSS, and JavaScript. | Windows has a large amount of IDEs that are available. Visual Studio code can be used for HTML, CSS, and JavaScript. Eclipse could be used for Java. It can also use Python IDLE for Python. | The official IDE for android is Android Studio and it can be used with Java, C++, and Kotlin. For Development on the Apple iOS Xcode is required and it supports C++, Java, Python, and a variety of other languages. Because of the need to develop for both Android and iOS, two development teams might be necessary, and this will also increase the cost. |

## 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**:

The recommended operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments is Linux. The first major advantage of using Linux is that it is free and open source. This means helps lower the cost of operations but it also means that if the company wants to make changes to the code, then it can. Additionally, Linux receives a lot of community support which can be useful in resolving problem or in making the system more efficient. Hiring a programmer that is familiar with Linux could also mean that the company could have the programmer design a user interface that is specifically designed for the company to use and will make managing the system easier.

Mobile operating platforms such as android or iOS are not well suited for use as servers and that lack of optimization will hold back the company’s ability to expand the Draw It or Lose it game. The Mac OS is not ideal, especially as a server, as it does not scale well into a larger environment. Additionally, Mac OS is more expensive than a Linux system. Windows is expensive, this includes the cost of the operating system, but because Windows uses more of the computer’s resources to run than more powerful hardware is needed which significantly raises the costs.

1. **Operating Systems Architectures**:

The hardware is the main part of any system’s architecture, this includes disks for storage, memory (RAM), the central processing unit, and any other hardware that may be necessary based on user demand. For example, if there are some graphical features that need to be rendered than a graphics card will be needed for the hardware. The Kernel is the core of the Linux operating platform. This is where the application programs in the system run. Additionally, this is the part of the architecture of Linux that handles the system’s hardware. The next part of the architecture is the System Libraries. These are needed for implementing the functionality of the operating system. The next section of the architecture is the Shell. This is an interface that is used between the user and the kernel, and its primary purpose is to take commands from the user and execute them according to the kernel’s functions. The shell can be a command-line interface where there are lines of commands that the user can input, or a graphical shell that uses a graphical user interface that is friendlier to use but runs slower.

1. **Storage Management**:

The company could use an offsite storage like Microsoft Azure or Amazon’s AWS (Amazon Web Services). These can both be used to store and retrieve files on command while still being compatible with the Linux operating platform that can be used on site. The downsides of this are that it will cost more to use these services, and the cost goes up the more they are used since they are pay as you go, which means that as Draw It or Lose It expands, it will cost more to use offsite services.

It is highly recommended that the company use its own onsite servers, since they can directly handle the servers and expand them as it is necessary while maintaining greater flexibility over the functions of their system. For onsite storage, it is recommended that the company use SSDs (solid state drives). They are faster than hard disk drives and they do not generate as much heat, making more efficient than hard disk drives. The only setback is that the cost of SSDs is higher than hard disk drives, but the cost is worth it for the better performance. SSDs are a piece of hardware that stores data, it can be used with Linux operating system.

1. **Memory Management**:

When Linux uses the memory that a system has it creates virtual memory. Virtual memory is a combination of RAM (Random Access Memory) and swap space, which is a section of the hard drive that is designated for use in case the is not enough RAM available. This is used to prevent any processes that are running from overlapping. By Doing this, disk fragmentation can be avoided. Additionally, virtual memory can be expanded beyond the physical RAM that is available, but it does sacrifice some efficiency. This means that even as more users use the Draw it or lose it software, the system will be able to keep up with each user and each game, since Linux will make sure that each game will have its own virtual memory space and that will help make sure that there is enough memory for each game that is being played. It is highly recommended that the software should be designed to allow for only a few images to be sent to users at a time, and thus lower the demand for more RAM. The lower the amount of data that is being sent to users, the lower the amount of RAM is necessary, thus lowering the demands that are placed on the system as a whole.

1. **Distributed Systems and Networks**:

The Draw It or Lose it software will be played over an internet connection. This means that multiple servers should be setup to run this software. The multiple servers mean that if one server were to experience some problems, then users that go online to run the software will be directed to a server that works, thus giving time for the company to fix the server that does not work, while still allowing their users to access and run the software. If the company has multiple locations, then it is highly recommended that the servers should be distributed to each location. This will mean that if one location experiences problems then another location can pick up the traffic, and it also means that users can be connected to the server that is closest to them, which is important for removing latency which can have an effect on a time-based game like Draw It or Lose It. Using a client-server architecture means that users can send requests to a REST API which can handle accepting user request and sending information back to users. When files are sent back to users it can be sent in the form of HTML, Python, PHP, JSON, or plain text. This means that it can be used to communicate with people using just about any operating system, from Windows and Mac OS to Android and iOS.

1. **Security**:

When users open the application on their end (the client side) they would be asked for their username and password. These credentials are then sent to the server that stores the user database to authenticate whether or not the username and password match up. This also involves passing a firewall that the database is hidden behind. If they do not, then the user will be prompted again for the information. If the authentication is successful then the user will receive privileges that are specific to their account. A unique feature of Linux that helps with security is that it is open source which means that the operators of the system in the company can inspect the code directly to find viruses and other malware that could affect the system. Additionally, Linux has a driver memory isolation which means that any process that is being run is not able to access any function or data that it does not have authorization for.