

Draw it or Lose it

# **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 | 11/20/22 | Shawn Murphy | Polished off iterators for game methods, implemented team and player id methods in gameservice.java |

**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 has asked for a web-based game that is compatible with multiple platforms which is based on their popular Android game, Draw It or Lose It. The games name with be based upon a popular game show in the 80s named Win, Lose, or Draw.

## [Design Constraints](#_2et92p0)

The game needs to run on multiple platforms with multiple teams of greater than 1 person.

Needs to check for unique team and game names.

Only one instance of the game may be opened at a time.

## [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 super class is labeled Entity, with all three classes Game, Team and player inheriting from it. The GameService references the Game class. While the Game class also references the Team class who itself has a reference to the Player class. This is done through aggregation. Along with these classes there is the ProgramDriver class and the SingletonTester class which has a use relationship with each other. Because of this, the program is allowed to have one game running with multiple teams consisting of multiple 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 provides the most security for hosting out of the options. However what comes with the complex security measures also comes a higher price point for hosting services. Along with this we get a command shell allowing for simpler server config, and an easy to use GUI. | Much more difficult to navigate however it is cheaper as it is very flexible. Linux also has a command shell which makes it much easier to perform server configuration while also providing more straightforward accessibility. | Comes with expensive servers and a more user oriented GUI. Provides a command prompt and boasts a much larger library of software compared to the other platforms. | Specifications will change from case to case. The specifications tend to be more favorable when looking at other devices. |
| **Client Side** | Mac tends to run at a higher price point as well as some knowledge of the platform may be required. GUI can need some practice to navigate easily. | High level of skill and knowledge is required to make use of this system. Very low cost to begin but again comes with a high gate of knowledge to run. | Cost of entry is higher than that of Linux but in most cases not as high as MAC. Does not take long to understand the system and is a much lower requirement of expertise than Linux. | Allows for an increased ability to access information however this flexibility comes with a cost of requiring more time taken to learn the platform and a high amount of knowledge and expertise is needed. Along with this it is more difficult to integrate other devices. |
| **Development Tools** | Some of the languages provided are HTML, CSS and JavaScript while also providing IDEs such as JavaScript, Python and Ruby. Some of the tools that are provided as well as PyCharm, Eclipse and Notepad++. There is always online tools as well. | Linux provides HTML, CSS and JavaScript languages as well as different libraries to back the languages. While also providing IDE’s such as Python, JavaScript, Ruby and PHP. Linux also carries tools such as Visual Studio, PyCharm, Eclipse and GitHub. | Again Windows provides HTML, CSS and JavaScript as it’s languages. And again it provides backing libraries for the languages while utilizing the tools of Eclipse, PyCharm, Visual Studio and a command prompt. The IDE’s given consist of HTML, C++, Python. | Similar to the other three, languages consist of HTML, CSS and JavaScript along with some backing libraries. Some of their IDE’s would consist of Python, PHP and C++. Some of the tools that can be utilized would include a command prompt, PyCharm, Eclipse and Visual Studio |

## 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**: Personally I would recommend the Windows server OS, this is because it provides greater results while also providing more software. Along with this and the minimum expertise required and the price point of it compared to other OS’s it shows how useful it is when you start utilizing it. The Windows server operating system is also designed to run on the hardware servers and is supported by many different server roles.
2. **Operating Systems Architectures**: Windows server operating system provides components to help with file management allowing the user to easily control and manage the systems memory. Along with this it also allows the developers to work and experiment with many different program languages.
3. **Storage Management**: Windows server OS provides superior memory oversight and allows for the application to be stored directly on the phone’s main memory. This allows the application to have a faster processing while loading. Not only this but with the increase in migration to cloud systems, windows also provides this option giving a large quantity of storage space.
4. **Memory Management**: Windows server operating system provides amazing memory management, as it provides virtual and physical address spaces with two to four gigabytes of memory. This is accompanied by very efficient runtime for applications.
5. **Distributed Systems and Networks**: While being slightly hampered by issues such as routing and congestion, windows server OS provides simple communication between each other and makes direction much clearer between work station. Not only this but it allows the use of a networking system, furthering how easily communication is within the OS.
6. **Security**: While providing account control settings which provide the ability to protect data going in and out of the system, there is also several VPN services that are compatible with the OS one included by Microsoft themselves. There is also built in anti-malware and anti-spyware.