

<Draw It or Lose It>

# **CS 230 Project Software Design Template**

Version 1.0

## 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 | 2/20/24 | Isaiah Ingraham | Building a multi-platform web-based game. Revising the design and domain of the model and updating the evaluation 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 multi-platform web-based game that is based on the current game “draw it or lose it”. The current game is only available on android, and they are looking to make it a web-based game as well. The client wants certain requirements to be met for the game application. Because of the requirements we will need to use the code from the Android app and adjust it, so it has the capabilities to run on multiple web browsers. We will use a Singleton pattern so that one instance of the game can exist in memory. We will also be using iterator patterns to make the game and team names unique.

## Requirements

* Each team has multiple players on it.
* Unique team names and game names to allow users to see what name is already in use.
* One game instance at a time.
* Have one or more teams included.

## [Design Constraints](#_2et92p0)

* Using the existing code in the Android app and making it work on web browser.
* Testing the app to make sure it is compatible with multiple platforms.
* Game and team names must be unique.
* Code to support multiple teams with multiple players per team.

## [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 diagram shows how the “Draw It or Lose It” game will be implemented. The diagram has seven classes that establish the environment that The Gaming Room has requested. The Entity class creates a relationship with Game, Team, and Player class, giving them the ability to get information from Entity (inheritance). The Game class holds a private list of Teams and abilities to. The Player class extends Entity to hold the information of the players in the game, each Team will have a private list of Players and the public method to add another player. Lastly, GameService will hold a private list of Games and can add a game instance and check if the game has a unique name. The GameService is a singleton class, meaning it only allows you to have a single instance of the game to exist at a time. To verify that there is not another instance of the same name we need a SingletonTester class to be used in the ProgramDriver class. One object-oriented principle used where inheritance, like Entity using the three subclasses of Game, Team, and 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** | +Easy accessibility/administration  +Good Tech Support plans  +less attacks due to fewer users  -less users so less information about the OS  -Not open source | +Easy to use.  +Customizable distro  -Learning curve/takes time  -Costly | +Easy to develop a server and administration.  +Has command prompt  -Not open source  -Less secure  -limited customization | +Code is hidden from the user.  +stores data often  -connection problems to internet  - Device specs vary from user to user |
| **Client Side** | +Easy to develop on a Mac.  + Works well with Apple devices  -Expensive  -Need to have knowledge of Mac | +Free  + Easy for people who use it, used often.  -Not usually the choice for playing games on | +used most in gaming.  +Updated more frequently, most used OS.  + little expertise needed to use.  -Not as secure | + Flexible for clients to see updates.  - less configurations than on a computer.  -Device brands react differently, |
| **Development Tools** | The coding languages are HTML, CSS, and JavaScript. Support front-end development. Other Tools are PyCharm, GitHub, Visual Studios and more. | The coding languages are HTML, CSS, and JavaScript. Support front-end and languages. Other Tools are JavaScript, Ruby, PHP, and Python. | The coding languages are HTML, CSS, and JavaScript. Support front-end development and languages. Other Tools are Eclipse, command prompt, PyCharm, Eclipse, and more. | The coding languages are HTML, CSS, and JavaScript. Support front-end development and languages. IDEs for programming are HTML, php, C++, and Python |

## 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 believe that the best option for operating platforms that will give The Gaming Room the ability to expand Draw It or Lose It the most would be Windows. Comparing the three platforms, Windows is the most used platform which gives the benefits of being able to deal with user requests and troubleshooting. “Windows OS is utilized by 95% of clients so the majority of the equipment merchants make drivers for windows.” (GfG, 2020). Also, Windows does well with web-based applications.
2. **Operating Systems Architectures**: Microsoft Windows is an OS developed and published by Microsoft that gives a large range of IDE’s which can be used to develop applications like Visual Studios and Eclipse.
3. **Storage Management**: Something that I use to keep my windows computer’s storage clean is Storage Sense. Storage Sense can be turned on in the settings and will clean all your temporary files and recycling bin over time. Saving your disk space for more important files.
4. **Memory Management**: Windows uses Storage Sense to help manage files on the computer’s hard drive. Storage Sense will also help Draw It or Lose It keep its photons and game players stored and managed more efficiently by keeping them organized and secured in the memory.
5. **Distributed Systems and Networks**: The distributed systems and networks can help with communication between the different platforms for Draw It or Lose It. The distributed system will give the games database and share it among the other users(players) and interact with one another over the network. This will allow multi-user interaction and gameplay between the different devices and platforms. Creating backup servers as a backup for when there are network outages to keep the gameplay going even during connectivity issues.
6. **Security**: Windows built- in security protection software is used to secure the users’ data and information. Although Windows does have a lack of security, you could always use another source to strengthen the security.

Works Cited:

GfG. (2020, December 28). *Advantages and disadvantages of windows operating system*. GeeksforGeeks. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-windows-operating-system/>

GeeksforGeeks. (2023, November 23). *What is linux operating system*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-to-linux-operating-system/>

Sharma, A. (2024, January 8). *Why you should not use mac OS? what is bad and good about it?*. TechWorm. <https://www.techworm.net/2018/12/mac-os-pros-cons.html>