

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

**Game Description** 3

[**Domain Model**](#_8h2ehzxfam4o)4

[**Evaluation**](#_2o15spng8stw)5

[**Recommendations**](#_m8aleynsvzvc)7

**1. Operating Platform** 7

**2. Operating Systems Architecture** 7

**3. Storage Management** 7

**4. Memory Management** 7

**5. Distributed Systems and Networks** 7

**6. Security** 8

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 10/07/22 | Adrian Sanchez | Included “Game Description”; expanded Table of Contents |

**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 that serves multiple platforms based on their current game “Draw It or Lose It”, which is currently available on Android. The Gaming Room wants the following **software requirements**:

* **A game will have the ability to have one or more teams involved** – Different game types: Single Player Mode (1 team scenario, player vs player), Team Play Mode (more than 1 team involved, team vs team).
* **Each team will have multiple players assigned to it** – A team will consist of 2 or more players. Will there be a maximum player count for a team?
* **Game and team names must be unique to allow users to check whether a name is in use when choosing a team name** – A lookup for game/team names should be created to solve this problem.
* **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.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

1. Will there be cross platform play for web-based and Android users to play the game together? The current game design is built for Android only and will take time to translate the game (GUI, client/server interaction, etc.) to web-based play. HTML5 & Javascript are supported in browser and mobile (to include Android), but we want to make sure your game runs seamlessly.
2. What browser would you like the game to be focused in first? (Chrome, Firefox, etc.) Just because the programming language is supported and portable to web, doesn’t mean it’s inherited in the game… The game will run in all browsers, but time will be required to debug and fix any issues that creep up from porting the game to web. There’s many different browser’s out there and all have different variant’s of the browser platform (desktop browser vs. mobile browser for instance). If development of your mobile game took a while, know that it will take longer to port to browser because there are many more nuances.

## [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.

**Game description**

The software application will render (draw) images from a library of stock images as clues to help players guess what the puzzle’s correct answer is. A single game consists of 4 rounds of play lasting one minute each (4 minutes total). Drawings are rendered at a steady rate and are fully complete at the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.

## 

## [Domain Model](#_8h2ehzxfam4o)

The ProgramDriver class contains the main() function and uses the SingletonTester class’s testSingleton() function. Entity class is a base class introduced to hold common attributes and behaviors. The Game, Team, and Player classes all inherit the Entity class. Game, Team, and Player all use id: long & name: String. The GameService class has Games, the Game class has Teams, & the Team class has Players (All association)

"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)

Outlined below is my evaluation of the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices. **For Server Side**: Evaluations are based on hosting a web-based software application. Considerations include each OS’s deployment method (server-based) and where the website will be hosted, as well as potential licensing costs to the client for server OS? **For Client Side**: Evaluations are based on cost, time, and expertise required to support the multiple types of clients. Considerations include required application development processes to ensure the application is compatible with all web browser platforms and mobile devices. **For Development Tools**: Evaluations are based on programming languages used for relevant OS’s and the tools that are used to build the software for deployment on each OS. Considerations involve impact to technical requirements on the development team as well as licensing costs related to development tools.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | * Apple discontinued macOS on 04/21/2022 * Server used to cost $20. * Server was notoriously bad | * Best OS for minimalist approach. * Hat Enterprise Linux is most reliable & popular Linux OS server * Cost $349 - $1,299 per year. * Free option with standard Linux OS if too costly. | * Microsoft Windows Servers has excellent file management system * Requires Windows software purchase * Requires server fees from $501 to $6,155. | * App hosting server between $70 -$320 per month * Cost ranges depending on content * Cost ranges depending on user traffic * Cost ranges depending on projected growth. |
| **Client Side** | * Mac is one of the mainstream OS’s. * Cost will not be very high comparatively | * Lots of expertise and time * Linux OS is minimal * Linux has compatibility issues * Need to hard code many fixes. * Cheapest/free | * Easy to hire for Windows client side. * Extremely integrated. * Integration time minimal * Cost is slightly more than Mac. | * Mobile devices are ubiquitous. * Added flexibility to clients & developers * Can see/make updates with phone access. * Requires dedicated time/expertise * Cost would make implementation harder. |
| **Development Tools** (available on all OS) | * Programming languages include HTML/CSS/JavaScript. * C/C++, Python, Java, C#, Ruby, etc. provide more functionality for complex games. * Resources/IDE’s include Visual Studio Code, PyCharm, Eclipse, & Github. | * Programming languages include HTML/CSS/JavaScript. * C/C++, Python, Java, C#, Ruby, etc. provide more functionality for complex games. * Resources/IDE’s include Visual Studio Code, PyCharm, Eclipse, & Github. | * Programming languages include HTML/CSS/JavaScript. * C/C++, Python, Java, C#, Ruby, etc. provide more functionality for complex games. * Resources/IDE’s include Visual Studio Code, PyCharm, Eclipse, & Github. | * Programming languages include HTML/CSS/JavaScript. * C/C++, Python, Java, C#, Ruby, etc. provide more functionality for complex games. * Resources/IDE’s include Visual Studio Code, PyCharm, Eclipse, & Github. |

## 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 suggest using the Windows OS for your platform since most gamers are on Windows OS platforms. Also, many people use the Windows OS as their main OS due to its online/professional support. Although there will be a higher cost to pay, the flexibility that Microsoft offers, because Draw It or Lose It will be web-based, will allow for better handling of user traffic, troubleshooting, and overall user-friendliness. These factors will aid in future player growth in Draw It or Lose It.

1. **Operating Systems Architecture**:

Windows OS provides a technically superior GUI when combining the user-friendly interface with it’s easy to use file management system and navigation tools (thanks to Windows NT) when compared to Linux, Mac, or Mobile. There are essentially 2 modes: User & Kernel Mode. User mode runs applications, while the kernel mode runs the core OS components. Windows also provides popular/powerful IDE’s such as Microsoft Visual Studio that has tons of support. Other IDEs would include Eclipse, NetBeans, PyCharm, Komodo, and many more that programmers are familiar enough with to troubleshoot any problems you have later down the line.

1. **Storage Management**:

Windows OS mostly manages storage through cloud-based storage but has availability for storage from local storage on a desktop to external storage arrays (dedicated storage hardware which has an array of HDDs or SSDs). There is also software available within the OS to cleanup recommended files, such as unused files, or files that can be placed into cloud storage so that they aren’t cluttering the physical storage.

1. **Memory Management**:

The Windows OS handles memory on a case-by-case basis. The memory is handled by 4KBs at a time called “pages”, both for physical and virtual memory. Applications memory will receive 2GB memory on a 32-bit Windows OS application and 8TB or application/kernel virtual address (VA) space for a 64-bit Windows OS. Depending on implementation of the Draw It or Lose It software, the Microsoft OS can help applications that make frequent allocations from the heap perform better by allowing the better process management of the memory in it via operations like HealAlloc and HeapFree.

1. **Distributed Systems and Networks**:

Although Windows has a lot of support behind the OS and support for developers, there is still going to be an uphill battle to get cross platform functionality when it comes to lag, connectivity, outages (blue screening), system updates, etc. These dependencies can be alleviated through Microsoft’s .NET framework which targets these dependencies to help add cross-platform support. These problems would be harder to deal with in the other OS. When trying to resolve these issues, the best way is for teams to communicate effectively the ongoing/recurring problems that arise.

1. **Security**:

Cyber security should be on the top of everyone’s mind when developing software for consumers/clients. Window Security is one of the best pre-packaged antivirus software programs available to come with an OS right out of the box. The antivirus software scans for malicious software (malware), viruses, and security threats continually in real-time to protect your servers and clients. Updates are downloaded regularly and automatically so you don’t have to search for them and your devices can stay protected. Nobody wants their information stolen or their accounts hacked into, and the Windows OS helps. However, it is still the developers duty to program the software with best practices and to ensure secure cryptologic measures are taken to ensure proper encryption of sensitive information.