

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 | 09/16/24 | Skylar Walker | Completed the executive summary, Design constraints, and Domain Model |
| 1.1 | 10/01/24 | Skylar Walker | Work on Development requirements chart |
| 1.2 | 10/03/24 | Skylar Walker | Work on Development requirements again |
| 1.3 | 10/16/24 | Skylar Walker | Work on recommendations |
| 1.3 | 10/18/24 | Skylar Walker | Finish recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming room staff wants us to set up an environment to run their game Draw It or Lose It. They need help with development. Program must run in a web-based platform with the ability to house multiple teams, players, and games at a time.

## [Design Constraints](#_2et92p0)

1. Having the game be made in a web-based environment means it has to be able to run on almost all browsers around the world such as Safari, chrome, and Firefox.
2. Since the program will run in a web window there will always be the effect of latency. We must keep in mind the impact of people’s input and getting the information from the players to the program and back to all the players.

## [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 game, team, and player class are all inherit from the entity class. GameService, Game, Team, and player all reference each other and the ProgramDriver class is where the main part of the program is run. The singletonTester lets us ensure there is only one instance of a game or team at a 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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Use Unix system  64-bit computing  Open-source development  Network authentication  Can have a high cost | Enhanced security  Open source  High scalability  Low cost of running servers | Widely used OP  Runs lots of different systems  Servers can be free  Higher security risk with the most popular OP. | Variability in device hardware  Oracle servers  Support for both android and IOS  High cost for oracle servers |
| **Client Side** | Mac has user friendly UI  The cost of running on apple can get high quickly. | Not all Linux UI is the same. Need to consider different experience for different users.  Longer time for development. | Windows licensing cost more than other options.  User friendly | Screen size  Touch screen UI needs considered in design |
| **Development Tools** | Xcode is a popular IDE for apple products.  C  C++  Java  node.js | Visual studio  PyCharm  C++  Python  JavaScript | Visual studio  C  C#  Java | Java  Kotlin  C++  Android studio  Xcode |

## 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 recommend windows for the operating platform for the gaming room. Widows is relatively cheap to work with as well as being the primary operating system used around the world (Statista, 2024). Windows is also easier to train people with given its vast popularity.
2. **Operating Systems Architectures**: Windows architecture lets applications use a layered design of a user mode and a kernel mode. The application can use the kernel to give the user the ability to have a graphical user interface for the game (Explore windows architecture).
3. **Storage Management**: Cloud storage would be my recommendation for this project. The expandability and customizability of it make it a good choice since you can choose how much storage you want and only pay for what you need. The cost of cloud storage over your own storage is better when you have to consider running out of space and needing to expand.
4. **Memory Management**: Windows has its own way of processing physical as well as virtual memory. The operating system uses virtual memory to tell the physical memory what to run. Windows can also use a process called disk paging where it will set aside a portion of its hard drive as additional physical memory if it happens to run out of RAM (Sinha, 2015).
5. **Distributed Systems and Networks**: I would recommend a platform such as unity that can be used on all operating systems. You would have to think about the cost per download that unity has but that is only after you reach a certain threshold of downloads. Also having a dedicated gaming server would be able to support a larger number of players at one time.
6. **Security**: Microsoft defender is windows main security when it comes to the operating system. The security on the storage side is based on what cloud storage system you decided to go with. Defender helps keep the information safe through its anti-malware. They also have the ability to run checks on the system for any vulnerabilities that could lead to data breaches (Chrisda, 2024).

References

Sinha, S. (2015, November 11). *Physical and Virtual Memory in Windows 10*. Microsoft. <https://answers.microsoft.com/en-us/windows/forum/all/physical-and-virtual-memory-in-windows-10/e36fb5bc-9ac8-49af-951c-e7d39b979938>

Chrisda. (2024, April 24). *What is Microsoft Defender for Business? - microsoft defender for business*. Microsoft Defender for Business | Microsoft Learn. <https://learn.microsoft.com/en-us/defender-business/mdb-overview>

Statista. (2024, September 24). *Global market share held by computer operating systems 2012-2024, by month*. <https://www.statista.com/statistics/268237/global-market-share-held-by-operating-systems-since-2009/>

Explore windows architecture – Training Microsoft Learn. <https://learn.microsoft.com/en-us/training/modules/explore-windows-architecture/>