

Draw It or Lose It

**CS 230 Project Software Design**

Liriel Hamilton 4/19/2025

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 4**](#_Toc115077324)

[**Evaluation 5-6**](#_Toc115077325)

[**Recommendations 7**](#_Toc115077326)-8

**Bibliography**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/22/2025 | Liriel Hamilton | In this revision I added the actual game name to the cover page and wrote out the three sections of Executive summary, Design constraints, and Domain model. |
| 2.0 | 04/6/2025 | Liriel Hamilton | In this revision I added the evaluation chart. |
| 3.0 | 4/19/2025 | Liriel Hamilton | In this revision I put in the recommendation section and some modifications to the table of contents. I also added a small Bibliography. |

**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, which is the new client, wants to create a web-based game called Draw it or lose it. They want this game to be able to be played on multiple platforms. At this time, it is only on Android. The game is very similar to the 80’s game show called Win, Lose or Draw, which was a team-based game. Instead of a player drawing each image and having others guess, it is instead generated from a computer database. There are four rounds and the teams are timed.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

* The Gaming room staff do not know how to set up a web-based version of the game and need help from us to design such.
* It needs the ability to have multiple teams play it.
* The game needs to keep at high performance no matter how many people are using it.
* A security system needs to be in place to protect users.
* Draw it or lose it, needs to be able to be played on multiple platforms, instead of just Android.
* Only one game can be played at any one 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 UML model has seven different classes that generate the diagram. The class Entity is added to make a parent class and Game, Player and Team are child classes of such. The UML model below shows this inheritance. The Game Service class is responsible for running the game and its different classes. It gets each game, team and player and also controls it going to the next one of each. The singleton tester is it to make sure that there is only one game playing at a time. The Program driver is the system to run the whole program.

**"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 or IOS created by the company Apple do have stable type platforms for different uses. It is very user friendly and developer friendly. A downside is that it does cost more than a lot of other systems with Apple being very big on everything being done with only their technology, so no third-party hardware or software. It is not a very Scalable system. Security on Mac is usually very good. | Linux is an inexpensive or free to use platform that is open source, however it is also not as user friendly as others and average users might find it too difficult to use.  It does offer many advantages for hosting things on the web. It does not always work good with working with windows programs. It is great for customization and making your own security and is very scalable. | Windows is fairly expensive; however, it is very scalable and compatible with a broad range of software. It is one of the main platforms used in computers these days. It does have a lot of security problems, and you do need to have good security software on systems with it. Third-party software works very well with this platform/ | Mobile devices are not as good as full computer platforms with web-based systems. Hardware is definitely more limited than computer-based platforms because of size constraints. Screens are also much smaller. Costs can vary depending on which mobile platform you are using, such as Android, IOS, Oracle, Google, etc. The main positive thing about mobile devices are their portability. |
| **Client Side** | Mac is a very easy to learn platform that many people these days have. Of course, the costs can be a major downside to anyone using Mac. Compatibility is also a major issue with Mac. | Because of its steep learning curve, it might not be a good choice for many clients. However, if they can learn it, the compatibility with other systems is a plus as well as the fact of it being free and open sourced. | Just like Mac it has a fairly high licensing cost, however its range of compatibility with other systems and software is a major plus. It is also a massively used platform these days and supports more games than other platforms. | The various different platforms for mobile will work differently than each other. Any application being made for it would need to have a concentration on it working with a Mobile type of platform. Ultimately it can be a very flexible system though. |
| **Development Tools** | Development tools include Xcode, JetBrains and eclipse. Common languages are JavaScript, Python, Java, CSS and HTML | Once again Java, JavaScript and Python as well as C and C++ are often used and IDE’s like JetBrains, Eclipse and IntelliJ IDEA. | All common programing languages can be used with such. These include Python, C, C++, Java, JavaScript, etc. IDEs include JetBrains, Eclipse, Visual studio and others. | Each mobile platform has its own IDE’s needed. Ios uses Swift code with Xcode and Android uses Java with Android studio. These are just a couple of examples. |

## 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**:

For Operating platforms, I would say for ease of use and to reach most users you should run concurrently on Windows, IOS and Android. All of these are fairly easy to learn. The only problem is that this will cost more than say using Linux. The problem with Linux is that only certain people actually use it, and it is much harder to learn. Windows would be the Platform that I would most suggest as it works with other systems well and has a wide array of toolsets. Also, Windows is probably the most widely used operating system of all.

1. **Operating Systems Architectures**:

I would set it up with the abilities of using windows 10 and up, as most Windows users have these operating systems. A lot of users have Windows 11 these days and so it has to have the ability to use the latest windows models. Windows also has a hybrid system of a User mode and a Kernal mode that the user has no access to. The Kernel mode runs the main systems in the background so that the user doesn’t have to worry about it.

1. **Storage Management**

For Storage I would say that the minimum that the user’s computer has to do is optimal. Having the files on a server-based cloud storage would be a much better idea and would be accessible to all. For the Users device a good amount of storage is always a plus and computers these days have a large amount of it, such as multiple terabytes for many of them.

1. **Memory Management**:

For memory management a paging virtual memory system would be Ideal so that it can keep up with the game. Luckily windows already has a system for this. Also, it would be ideal for the user’s computer to have as large of a memory chip as possible. A good Processer is also a plus these days.

1. **Distributed Systems and Networks**:

For the Distributed systems and Networks, a cross-platform environment would be the way you want to go with this. Such will make it smoother to run the whole game without the user having to be a computer genius to run things. The system will also have to run at peak performance so that delays and other issues don’t happen as much, so constant updating is a must.

1. **Security**:

For security I would use F-Secure Elements (Brame, 2022) from looking at a list of the 10 best security apps. It is good with mobile devices; it is excellent with customs and policy management; it is great with its detection performance, and it has bundled patch management. It also has a complete online security package that protects from scammer and protects your Identity and keeps your privacy.

Brame, D. (2022, January 13). *The Best Hosted Endpoint Protection and Security Software*. Retrieved from PCmag.com: https://www.pcmag.com/picks/the-best-hosted-endpoint-protection-and-security-software?test\_uuid=01nEKitrPAQmw6jRcSPVArQ&test\_variant=A