

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 | 11/23/2020 | Chase Patten | Added date and author name |
| 2.0 | 11/25/2020 | Chase Patten | Added Executive Summary, Design Constraints and Domain Model |
| 3.0 | 12/06/2020 | Chase Patten | Added client side, server side and development tools |
| 4.0 | 12/14/2020 | Chase Patten | Added my recommendations for the project |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room is looking to create a web-based game that can allow for one team or multiple teams to participate in a game at once. This game will be like an 80’s tv show called *Win, Lose, or Draw.* Instead of players drawing the images and having their teammates guess the image, they will be using a program that will render and display the images for them.

## [Design Constraints](#_2et92p0)

Some of the design constraints that we will have to deal with are:

* Web-based application
* Will be programmed in Java
* Each team must have the capability to have multiple members. Each team also needs to have the ability to have a unique team name.
* The application will also search to see if a team name is not already taken upon name registration.
* The game will render images from a rather large library of drawings as clues. Each game will consist of four rounds of play each one lasting one minute each.
* The drawings are rendered and should be fully rendered for the user at the 30-second mark.
* If the team does not guess the puzzle before the remaining thirty seconds expires, then the remaining teams will have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.
* With a web-based game that tells me it is running on a browser. So, to me this should mean that the operating system running the web browser is relatively unimportant.

## [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 classes that are define in the UML diagram and their uses are as follows:

* Program Driver (Uses the other classes in order to run the program)
* SingletonTester (Used to test the Singleton class to verify it is working correctly)
* GameService (Used to operate the game service itself)
* Game (Here is where the game itself is ran)
* Team (Here is where the teams are built and set up and the information stored)
* Player (A single class used to keep track of the players as a single object)
* Entity (Game, Team and Player use this class as their base class as all three are also entities)
* The ProgramDriver uses the SingletonTester
* GameService, Game, Team, and Player all have an association of 0…\* This is because each association between the classes as shown can have 0 or many possible associations.

****

## [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** | The cost of developing server side for Mac can be a bit more expensive. We will be better off security wise as Mac products are not targeted as often for cyber-attack. The hardware however is going to be more expensive as all the hardware is proprietary hardware from Apple. Apple does have a support model that we can utilize in case of troubleshooting. | Linux is a completely free OS to use for both personal and commercially available for free or with support depending on flavor and the organization running the Linux distribution you choose. Linux also allows you access to many various browsers used by consumers. | With Windows you have a far greater access to different hardware form factors and types. With Windows you have access to some of the major browsers that clients use such as, Firefox, IE, Microsoft Edge and Google Chrome. Windows also comes with support for server-side products if we need it. | The application itself can be hosted somewhere else, whether through IoS or Windows and the mobile device will access the server from the phone to play the game. This can save resources on the user’s phone but, it will increase the resource requirements server side. |
| **Client Side** | The pros of going with Mac is that everything from the OS and the hardware it runs on is proprietary and we can get help from Apple if need be. However, that is also one of its cons. As we will have to pick hardware and software directly through Apple in order to work on their systems. We do not get the freedom of choosing what hardware and software to use. With Mac we will only be able to use the APFS file system. | The cost of developing on Linux is cheaper than both Windows and Mac as there are so many flavors of Linux that are free to use. We will need more specific expertise from those who know how to develop on Linux based systems. The time involved for the project may increase due to Linux’s ambiguity and if we choose a flavor that does not come with paid technical support. | The cost for developing on a Windows system will be less than Mac. Seeing as how Windows is the biggest OS used in the world it should be much easier to develop for in terms of cost, and effort and expertise needed. With more people understanding Windows, the easier it should be to develop for. With Windows we will only be able to use the NTFS file system. | For developing on mobile platforms, the cost, time and expertise needed will depend on which mobile device we develop for. If we develop for iOs phones, then we will need Apple’s permission to develop first which can be costly and time consuming. If we develop for Android based phones the cost will be less and we will be able to put it on the app store much easier. |
| **Development Tools** | We will need to develop in Objective-C. We can used Dreamweaver for Mac or we could use CodeWarrior Pro, which would cost us a fee or, we could use other IDE’s such as Roaster 3.0 or Café 1.5. | We can use a whole host of IDE’s on Linux systems. It would just depend on which languages we use. If we use Objective-C we can use Project Center or KDevelop. If we use C we could use NetBeans or even CLion. We could also develop the application with Java using Eclipse. | On Windows we could program the application in Java, Objective-C, or even C++. The IDE’s we could use are KDevelop, Clion, and Eclipse respectively. | We could use Eclipse Ide for Android as well as Visual Studio and Android Studio for Android. For iOs we could use Xcode or App Code. The languages we can use are similar to the other platforms as Objective-C, C, C++ or even Java. |

## 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**: If they want to make sure that their game can be seen and played by as many users as possible than I would recommend them to develop for iOS and Android based mobile devices. I would also recommend programming it for Windows based devices as well. This also means for tablets as well. Both operating systems support tablets and the tablets and smart phones will make it easier for a team-based game played with other to be played easier.
2. **Operating Systems Architectures**: Windows comes in a x32 and x64 bit architectures. This means that 32 or 64 bits of data can be brought into the processor at a time. We will have to host the application on a server where it can be running for the user to save the user resources on their local device. This means we will need a server operating system to run on the server itself.
3. **Storage Management**: We can run the storage required for the game with two hard drives. The first hard drive can hold the OS and the game itself that is running. While the second hard drive is used for parity. This can mean that if the server goes down then we can reboot it with the second disk and save time and money.
4. **Memory Management**: I would recommend a x64 bit architecture with about 256 gigs of RAM. This allows for more than enough space to run the game, host many players, and run the rest of the services needed to run the server. Also, while using a server architecture of Windows 10 Enterprise Edition we can have a max of 6 TB of RAM. So, we have more than enough space to upgrade the server’s RAM as needed.
5. **Distributed Systems and Networks**: We could distribute the systems required to run the game from multiple locations. This would allow us to distribute resources effectively and compartmentalize systems in case of failure. We could use a scaled down version of what Massively Multiplayer Online Games use.
6. **Security**: With Windows based platforms, it can be difficult to maintain security. Especially if the application is written in C++ due to various issues with the programming such as the use of nodes in memory. But, with programming for Android and iOS we get a much more secure set up. Especially with iOS as there are not as many security concerns with Objective-C as C++ for Windows. Also, the global popularity of Android makes it more of a target for criminals. While iOS is more secure overall this does not make it invulnerable to exploitation. We can attempt to make sure that the game comes with basic hacking and exploitation defenses so the game can be as fair as possible. We also need to keep in mind of the ever-changing nature of security and should make plans for post launch security patches.