

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 | 01/17/2022 | Jaelyn Sloan | Initial version of the document |

**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)

Creating a web-based game based off of an existing Android game application, Draw It or Lose It. The web-based version of the game will have the ability to have one or more teams involved, with multiple players assigned to each team. The game and team names that the players enter must be unique because only one instance of the game can run at a time.

## [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.>

Technical constraints:

* Operating systems should be for Windows, MAC, Linux
* Programming language
* The Gaming Room will host the cloud environment for the game
* Will be using the framework of the Android game application

Business constraints:

* No budget or schedule was given
* Licensing

## [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)

Child classes Game, Team, and Player all inherit from parent superclass Entity. These child classes all share the same properties that they get from the Entity class. Each class contains private objects, such as “id” and “name” in the Entity class which can only be accessed within the Entity class. The Entity class and GameService class both contain getter methods to allow for those private objects to be seen by the user, but not changed.

**"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** | Offers a server-based deployment method where the website will be hosted? YES  Advantages:  Upgradable  Has many options for web hosting requirements.  Disadvantages:  Expensive licensing | Offers a server-based deployment method where the website will be hosted? YES  Advantages:  Very secure/catches security breaches early on  Free/open source  Disadvantages:  Difficult to find supporting apps that meet host requirements. | Offers a server-based deployment method where the website will be hosted? YES  Advantages:  Most common OS for web hosting.  Has the most software available  Disadvantages:  Virus susceptibility  Licensing can be expensive | Offers a server-based deployment method where the website will be hosted? YES  Advantages:  Portability  Fast and responsive  Easy to install  Disadvantages:  Not secure  Not much software available  Licensing can be expensive |
| **Client Side** | Common OS so most developers can navigate easily with little expertise and time. Comes with Mac PCs, so no extra cost. | Requires much more time and expertise to learn and use. Free, so very cost friendly. | Common OS so most developers can navigate easily with little expertise and time. Comes with most PCs, so no extra cost. | May be slightly more time consuming and take more expertise to navigate. Many different OSs for mobile devices, but the device comes with it, so cost friendly. |
| **Development Tools** | XCode IDE recommended;  Swift, Java, Python, C++, Notepad, Ruby, AppleScript, VisualStudio  XCode is designed specifically for apple developers but is simple enough for beginner developers to use. XCode itself is free to use, but there is a yearly fee in order to publish applications.  May need a specialized development team familiar with Apple development. | Eclipse recommended;  Java, Python, C++.  PyCharm, VisualStudio  Eclipse is a free, open-source platform for programming languages. Easy and free for development team to use.  It should not be necessary to use multiple development teams. | VisualStudio recommended;  HTML, CSS, Java, Python, C++.  PyCharm, Eclipse.  Notepad  VisualStudio is a free IDE that supports multiple languages but is preferred to use with Java. VS is free to use.  It should not be necessary to use multiple development teams. | Android studio, XCode, Swift, Java, Python, C++  See MAC development tools for XCode;  Android Studio is and IDE specifically used for android app development. There is a one time $25.00 charge, so it is affordable.  May need a specialized development team familiar with Apple and Android development |

## 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**: The first operating system that should be used to develop the game is Windows because it is the most common and most easily navigated OS. Windows also has the most software available and is pretty cost effective since it comes on most PCs. The next best operating system to use would be Linux. Linux is open source, has very strong security features, and has multiuser/multiprogramming features. Using MAC or a mobile operating system would require more expertise and would not be as cost effective as using Windows or Linux.
2. **Operating Systems Architectures**: Windows has two modes: a user mode and a protected mode called Kernel mode. Kernel mode gives the programmer complete and unrestricted access to underlying hardware. This allows the programmer to execute any CPU instruction and reference any memory address. User mode restricts access to critical areas of the operating system in which they do not need access to. Windows also allows applications to show on the Graphical User Interface which allows for simple and straightforward use of those applications with little to no experience.

Linux also has a similar architecture consisting of a kernel mode and a shell. The shell is an interface between a user and the kernel that executes underlying functions. The user is able to use a graphical interface or a command line interface to perform tasks. Linux is also open source and multiuser/multiprogramming, meaning that multiple developers may work on the project at once.

1. **Storage Management**: Windows comes with a storage management system called storage sense which allows the user to manage their storage on a hard drive. Storage sense automatically frees up space by deleting items on the hard drive that are not needed, such as temporary files or items in the Recycle bin. Storage sense is also able to connect with OneDrive and perform the same tasks in OneDrive locations.

Linux uses a filesystem to organize and arrange files in the system. The user is also able to use a Logical Volume Manager (LVM) to create pools of storage and add hard drive space to those pools when necessary. It is more complicated to manage storage in Linux because it is not an automatic task that is done by the OS.

1. **Memory Management**: The memory allocation on Windows allows for easy access to folders and libraries that may contain the application files. It will be necessary to keep the project together in one location. Windows allows you to create folders in a chosen location to keep the files together.

Linux uses a hierarchical file system to structure the way files are arranged. This allows for grouping of files so that the user only sees files that are needed.

1. **Distributed Systems and Networks**: Network based games require some sort of database that can be shared with multiple users at once so that each user can interact with the game at the same time. Using Windows or Linux, the developer can create a database that allows for a network of gameplay that users can access at once. The broker pattern can be used in any operating system to allow for communication between server and client. This pattern uses a middleman which is essential in receiving requests and then performing the desired request from the client or server.
2. **Security**: Windows 10 and 11 come with Windows Defender. Windows Defender automatically detects virus and threats and stops them before they can cause damage. This runs in real time, so there is no need to manually go in and scan occasionally. Windows Defender is said to detect 99% of malware attacks in real time.

Linux is the most secure of any OS because it is open source, giving unlimited access to members which means that any security flaw will be noticed and fixed almost immediately. Linux also uses a very strict user privilege model where users have low access rights and must request permission to do certain tasks, which makes it much harder to spread malware.