

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 | 12/11/2021 | Joaquin Esguerra Jr | Software Design Blueprint |

**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 is looking to develop a web-based game that can function on multiple platforms. The title of the game will be “Draw It or Lose It” and Android is currently the only system to play it on. The game consists of multiple teams, holding multiple people, competing in timed rounds. For functionality, each of the games and team names must be unalike. The Gaming Room has a large library of stock images that will be used for the game, but the current employees do not have the know how to build the environment needed.

## [Design Constraints](#_2et92p0)

•  Development Kits differ from each platform

•  The game must work with different platforms

•  Game, player, and team names must be different

•  A sole game can exist in each period

•  Established communication link for alerts

## [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)

All classes inherit information from the created Entity class. The UML diagram illustrates this inheritance. All classes will utilize common references like “name” and “id”. When looking at relationships, Team and Player are “has a” types. While Game holds a Team and GameService holds Games, when using UML, it’s called aggregation (HAS A). When a use “has a” it describes having an instance of one class and reference to another class. When reviewing the diagram, GameService has a reference of Games, Games a reference of Teams, and Team a reference of Player.

**"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** | When considering Mac OS, a server is offered and available. The Mac OS server seems relatively inexpensive at $19.99 based on my findings from apple.com. Although Mac has gained popularity, through the power it brings to business, through further research, it seems as though Windows and Linux remain superior. | Linux brings another element to the server-side requirements. Its distributions have many server capabilities. Some highlights included the open-source environment along with an up-front low cost. A cost could become present when looking for experience with running a Linux server. | Windows seems to be the most used operating system known. Cost could be presented as a factor for the server-side but finding experienced users could offset initial cost for a fully functioning platform. | Compared to computers, mobile devices are not equipped with the same power to hosting a server. Regarding options, Android devices come the closes to capability and provide a low-cost alternative that can fit right in your pocket. |
| **Client Side** | With Mac operating system not being open source, the cost would be comparable to Windows. The primary factor when considering Mac would come down to experience. Non-experienced Mac users would need some time to familiarize with the system. | With Linux being open-source cost would not be a major factor. Like Mac, but even greater, experience would become a greater need. Linux is not only an unwieldy used platform, but even to a novice user, challenges are still presented with Linux. | Being non-open-source cost is an aspect for the client-side. Like the other platforms time and experience comes into play which too contributes to cost. An experienced Windows user would get right to work and a non would need some time to develop. | Mobile devices to not generate a huge factor in terms of cost. Most mobile devices are user friendly even if the user is not familiar with the platform. The ability to run specific apps on the client-side would not necessarily come down to experience, but the operating system. Some BYOD (bring your own devices) have better functionality than others depending on the platform being used. |
| **Development Tools** | Swift is a powerful and intuitive programming language with Mac OS. Swift also offers modern features that could be attractive to developers. When considering development tools there are several quality IDEs for developers. Atom, AppCode, CodeRunner, VS Code, and Xcode just to name a few. | Linux users experienced or not will benefit from great developer tools. The right tools and feature can lead to quality coding. There are several IDEs one could find success with but some of the best through research include Sublime, Atom, Brackets, Eclipse, Kate, Gedit, Greany, and the Bluefish Editor | There are several IDEs that can be used for Windows. Microsoft primarily uses Visual Studies. Eclipse, IntelliJ, PyCharm are also some others. IntelliJ offers IDEs for many languages and has great features for the intermediate to novice developer. | Numerous applications can be created using both Swift and Android. These languages can be run on various machines. Supported languages include JavaSript, HTML, and CSS which all tie into frontend management. Java, Python, and Ruby also can play a role when looking at general languages to use. |

## 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 recommend The Gaming Room use windows as a starting point. Windows has more software available and with minimum expertise and cost to get the various projects up and running. Windows is a widely used operating platform and as previously noted, individuals’ knowledge of its systems is widely known. Multiple IDEs will come in handy also.
2. **Operating Systems Architectures**: The Windows NT kernel is a hybrid kernel; the architecture comprises a simple kernel, hardware abstraction layer (HAL), drivers, and a range of services (collectively named Executive), which all exist in kernel mode. The Gaming Room can utilize the power of Windows for “Draw It or Lose It” to have GUI/Window set up, routes to memory and various other crucial processes that will make “Draw It or Lose It” run without effecting the overall operating process of the windows platform.
3. **Storage Management**: Designed to be a hub connecting all your data management features, Windows 10 Storage menu has many features allowing you to review and alter files quickly and programs to reorganize and open additional space. The features are known as storage sense and disk management. These maintenance features open up space by easily helping to clean up unnecessary files. While this feature is extremely convenient, it also means you can mess up or lose your data very easily if you don’t know what you are doing. The gaming room must understand the positives and the negatives and teach and train to combat both.
4. **Memory Management**: Windows provides a resource describing Windows Operating System Memory Management on their Windows Development Desktop technologies site that could benefit the game and The Gaming room future projects. Each process on 32-bit Microsoft Windows has its own virtual address space that enables addressing up to 4 gigabytes of memory. Each process on 64-bit Windows has a virtual address space of 8 terabytes. All threads of a process can access its virtual address space. However, threads cannot access memory that belongs to another process, which protects a process from being corrupted by another process. The Gaming Room will utilize Windows memory to build a database for the “Draw It or lose It” game image files for accessibility.
5. **Distributed Systems and Networks**: Due to the differences of operating systems, The Gaming room will need a cross platform creation. A type of IDE that can run on any device. Once created, the game can easily be exported into web, iOS, Android, and many more options that will allow cross play. Strong servers will be needed to prevent connection issues and be able to support large volumes of players. Since “Draw It or Lose It” allows for multiple players in a single game, the need of a strong server will be crucial.
6. **Security**: Windows 10 and 11 include Windows Security, which provides the latest antivirus protection. The Gaming room devices will be actively protected from the moment Windows is started. Windows Security continually scans for malware (malicious software), viruses, and security threats. In addition to this real-time protection, updates are downloaded automatically to help keep devices safe and protect it from threats. The security is a type of Defender and it works to encrypt all data that is sent back and forth from the server to the game by utilizing various encryption methods.