

Draw it or Lose it

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 05/20/2024 | Buxton McCaslin | Initial draft for The Gaming Room’s Draw it or Lose it |
| 1.1 | 06/3/2024 | Buxton McCaslin | Added details for evaluation section |
| 1.2 | 06/14/2024 | Buxton McCaslin | Added details for recommendations section |

**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 requesting a multi-platform web game for their Android mobile application, Draw it or Lose it. The game centers around multiple teams guessing a puzzle with images from a drawing library. Consisting of four rounds, each team and player have their own unique identities to prevent conflicts.

## Requirements

Requirements for this gaming environment will need to have the ability to have multiple teams with multiple players in each team. Additionally, since each team and player’s names must be unique, the environment must check whether a name is already in use. Lastly, there can only be one instance of the game existing in memory at any time.

## [Design Constraints](#_2et92p0)

Since every team and player’s name must be unique, in addition to there only being one instance of the game, the identifiers must all be different and only occur once. A singleton pattern must be adopted for object creation to accommodate this and an iterator pattern to avoid conflict between teams and players. Additionally, since the original game is an Android application, it will need to be compatible with web-based Java. Furthermore, the web application will need to be able to handle a large scale of visitors at the same time while providing a smooth and user-friendly interface.

## [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 diagram provided below shows the relationships and details of classes of the Gaming Room’s Draw it or Lose it software. The diagram consists of a main driver class that initiates the creation of a game, teams, and players. From there, the game service class represents the game session and holds unique and private attributes to start a game set up. Next, the game service class will then add the game and deposit it into the game list. After the game is added, a team will be added and placed into the teams list. Lastly, the players will be added and placed to the players list. The game, player, and team classes are subclasses of the entity class. The entity class holds everything together and has the default constructors.

The UML diagram demonstrates multiple object-oriented programming principles by way of inheritance, encapsulation, and abstraction. In the entity class, inheritance is used in the extension of the constructors. Whereas encapsulation and abstraction are shown in the methods used to add teams, players, and games to lists. Association is also apparent throughout the diagram as the classes and attributes interact with each other.

**"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** | MacOS is a well-known powerhouse of server-based environments. Having a MacOS server is advantageous for mac clients as it fully supports Mac applications in addition to an intuitive graphic interface. Additionally, Mac has a very robust anti-malware program for security. Not only does it protect against spyware, adware, and worms. Moreover, the flexible mac terminal also allows for configuration of the server so that accesses and changes are secure. The disadvantages are that having a Mac server is expensive to maintain and that it can become cumbersome for larger enterprises that rely on third party programs and customizations. There are also higher licensing costs due to only being able to purchase Apple products. | Linux is one of the more popular servers for web hosting. Some major advantages of having a Linux server is that it is free and open source. Deployment and tools would be minimal cost to the organization. Additionally, optimizing security on a Linux server would be incredibly beneficial as Linux servers are very customizable and secure. The application’s security could be customized and implemented as needed. Linux is also less prone to cyber threats. Lastly, Linux servers also support a wide variety of languages such as python, ruby, and PHP for example. However, the cons of Linux is that it can be difficult to grasp as it has a strong learning curve and is not the most user-friendly. Additionally, some software or applications have limited support, and it is harder to find drivers or applications that support web hosting needs. In that case, there can be migration issues from Windows to Linux. | Windows is one of the more popular OS along with Mac and has an extensive library of software that offers support for applications and third-party software. Windows is also widely compatible and is user-friendly. The interface is intuitive to use for most and well-known languages databases like MySQL and python are compatible. However, like Mac, since it is a proprietary OS, it is more expensive to maintain with licensing fees. Moreover, Windows is generally susceptible to cyber threats like viruses due to poor technical support. | Mobile devices inherently are accessible by a large user base and are user friendly. Both android and iOS systems have seamless interfaces and user experiences. Both have the potential to reach One of the advantages of mobile devices is that the software is less susceptible to cyber threats than other operating systems. Additionally, there are development tools available for both iOS and android systems. Some disadvantages of mobile devices are that there is not many companies that offer web servers for mobile devices. Both iOS and Android do not require server-based deployment but can be hosted on a server through backend. It can also be difficult to maintain support along older versions of Android and iOS systems. iOS, like Mac, has string guidelines for applications which offers less opportunity for customization. |
| **Client Side** | Software development considerations for Mac is that there is a moderate need of expertise and time for supporting multiple types of clients. While integration between Apply devices is straightforward, expertise is needed to navigate the OS has there are often long gaps between hardware updates. There is considerably less time involved as updates are less often and the OS is good at multitasking. Cost wise, the initial set up of the Mac system is expensive and some software available for Mac have monthly fees. | Cost-wise, Linux might be the most cost effective as it is an open-source system. However, as it has a strong learning curve, there is a significant increase in time and expertise required. Additionally, since there is not much tech support and there is a lack of patches, time wise, there are more bugs that need to be worked through. | As for Windows, due to the large library of applications and hardware, there is less time and expertise required. Many applications are compatible and thus streamlines the process for development. However, due to constant updates and functionality inconsistency to malware, there might be more of a time focus on security. As with Mac, there are more significant cost associated with licensing fees as well. | Mobile devices are superior in cost effectiveness as it translates to a larger customer base. However, on the same coin, since there is such a variety of hardware, updates and application maintenance might require greater expertise and time to resolve. Additionally, since both Android and iOS have their own programming languages, development of the software could also require more time and expertise. |
| **Development Tools** | Relevant programming languages for Mac include HTML, CSS, JavaScript, Java and Swift. Swift as an open-source language created by Apple is one of the more inherently compatible languages for software development on both iOS and Mac systems. IDE’s can include Java, Python, and Ruby. Development tools for building software on Mac systems include Visual Studio, GitHub, PyCharm and Netbeans. | Relevant programming languages for Linux include C, C++, Python, and Java. As for IDE’s, similar to Mac, they also include Java, Python, and Ruby. As for development tools, there is a large variety available open source to include Gedit, Vim, Netbeans, and Bluefish. | Windows supports a wide variety of programming languages. Most relevant programming languages include C++, Python, JavaScript, Ruby and Go. Similar to Mac and Linux, Windows can use IDEs and development tools like Visual Studio, NetBeans, and Notepad++. | As for mobile devices, relevant programming languages vary by hardware, Apple or Android etc. However, Java, JavaScript and C++ are the most widely used and cross compatible. For Apple specifically, using their programming language Swift might provide inherent benefits to using the native language. IDEs and development tools for mobile devices include Unity, Godot, GitHub, Visual Studio and Unreal Engine. |

## 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 the operating platform, I would recommend Windows to expand the game Draw It or Lose it. Not only is it user friendly, but it also is easier to learn. It doesn’t require as much expertise for developers to utilize. Additionally, Windows has an extensive library for software support with plenty of IDEs. While there might be platform specific areas of the code, Windows supports a variety of programming languages and libraries that can be compatible and used to an extent with other operating platforms. Lastly, Windows has the largest market share and the most users. This would be advantageous for The Gaming Room to target the most popular platform first.
2. **Operating Systems Architectures**: At the basic level, the Windows system can be divided into three layers. The first layer is comprised of the kernel and other low-level drivers and is considered the kernel mode. This is the core of the operating system and is where the interaction between hardware and computer processes are managed. Next, building up from there is the system services layer or shell layer. At this layer, without user interaction, deploys and communicates tasks with other software components in the system. Then, the top layer is the application layer. This is considered the user mode because applications are initiated by the user. It is comprised of applications from both the Microsoft store and traditional Windows desktop applications such as PowerPoint and Word.
3. **Storage Management**: When it comes to storage management, Windows has an integrated feature called storage sense. This tool helps in file management organization and reduces unnecessary clutter. In addition, storage sense also allows the user to dictate save locations. In turn, this provides easier file creation for larger projects. Moreover, I would recommend utilizing Windows cloud storage. With practically unlimited space, the gaming application would have increased scalability and resource management.
4. **Memory Management**: Windows memory management involves allocating both virtual and physical space. The physical space refers to the hardware component, RAM, that stores data actively being used while virtual space refers to address space for applications. By having two memory spaces, there is greater flexibility and capability to process large programs smoothly. The maximum possible memory space is 16 terabytes for Windows 64-bit hardware. With the game application requiring a large database of images, this memory management method is ideal.
5. **Distributed Systems and Networks**: For Draw It or Lose It to communicate between various platforms, implementing RESTful APIs are recommended. With RESTful APIs, sub -services can run independently and communicate with each other. This allows devices to interact with the application’s backend and exchange data with various platforms. Additionally, it is best to use a Windows IDE that is also compatible with other platforms to increase overall coding compatibility. In consideration of component dependency, dedicated servers or cloud servers are recommended to reduce the risk of issues arising. Dedicated servers are more stable and customizable to balance the game application load. There are also many cloud platforms that provide cloud servers for game applications with low cost and additional tools for scalability.
6. **Security**: Security of the application is of the upmost importance and while Windows is not known to be the most secure of operating platforms, there are options and resources to strength security. To begin with, Windows does have inherent security protection software that monitors in real time for malware, viruses, and other cyberthreats. However, it is recommended that within the application, measures to ensure best security practices be implemented such as access controls, user authentication, and encryption be implemented. Additionally, routine scanning for vulnerabilities and possible weak points and accompanying updates should be performed. Moreover, user data could be stored within servers, cloud or otherwise, as they provide more robust security.