

Draw it or Lose it

# **CS 230 Project Software Design Template**

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 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/15/2023 | Jessica Gonzales | To enable simultaneous team play functionality, data administration should be implemented to ensure distinct username titles among players, guaranteeing a single instance of the game |
|  |  |  |  |

**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 Company intends to design a cross-platform web game named "Draw It or Lose It." Currently, the game is exclusively accessible on the android platform. Its objective is to engage multiple teams, each comprising several participants, in a four-round battle. Each round lasts for one minute. Teams will make guesses on various randomly selected pictures from an image library until their time expires. In case a team fails to provide an answer, the opposing team will be given 15 seconds to respond before time runs out.

## 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)

When developing this game, several design constraints need to be considered. Firstly, the game requires the participation of multiple teams, each consisting of multiple individuals. Additionally, to ensure uniqueness, each team must have a distinct name that can be checked by the program for availability. Moreover, the game should be limited to running a single instance at any given time. Furthermore, it is essential for the game to be compatible with multiple platforms. Currently, it is only available on the Android platform, but the goal is to make it accessible on various devices, including Windows, Apple, and Linux. Therefore, it will be necessary to devise a strategy to modify the code and adapt it to function effectively across these platforms while meeting the aforementioned requirements.

## [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)

Utilizing an "entity" concept would be a good approach as it establishes a connection among the game, team, and player classes. This relationship enables them to inherit or acquire essential information from the entity class. This inheritance relationship is clearly visible in the UML diagram. Consequently, common attributes such as name and id will be shared across these classes, making entity the superclass.

**"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** | Expensive: The platform is costly but offers an excellent graphical user interface (GUI) that ensures ease of use for both developers and other users. Additionally, MAC provides a flexible workstation that allows for easy accessibility and server configuration. | Cost-effective: While the platform is cost-effective, it may present challenges in terms of navigation. Similar to MAC, Linux offers a command shell that simplifies server configuration and accessibility. | Costly: Similar to MAC, this platform is expensive, but it provides a user-friendly environment. Windows offers a wide range of available software options and features a command prompt for executing commands. | Inexpensive: Mobile devices vary in specifications, including Android, iOS, and Windows Phone. Developing a game that is compatible with most or all mobile platforms poses challenges due to the diverse nature of these platforms. |
| **Client Side** | The level of effort required to support MAC users falls within the average range. Adequate skills, similar to those needed for Windows OS, are necessary. However, it's worth noting that MAC is more expensive compared to Linux. | Supporting users of Linux OS requires maximum time and expertise. Mastery of Linux is crucial in order to navigate and assist users effectively. On the positive side, the cost associated with Linux is minimal. | Supporting Windows OS users typically requires minimal time and proficiency. The cost is relatively similar to that of MAC and OS. Therefore, providing assistance to Windows users can be accomplished with relative ease. | Supporting diverse mobile device users presents a significant challenge. Various mobile operating systems require maximum time and skills to ensure compatibility across different devices. The unique nature of mobile platforms makes support more demanding compared to other devices. |
| **Development Tools** | In the realm of MAC OS, developers commonly rely on popular programming languages and integrated development environments (IDEs) such as JavaScript, CSS, Python, and HTML, among others. They also utilize a range of tools and libraries like Eclipse and Visual Studio to support their development needs. | In the world of programming, it is customary to use languages such as Ruby on Rails, Java, Python, CSS, JavaScript, and HTML, among others. Developers rely on various development tools such as Node.js, Visual Studio, GitHub, Docker, and command prompt for efficient coding. | Developers often resort to customary programming languages like Ruby on Rails, Java, Python, C++, C#, JavaScript, and HTML, among others. Their preferred development tools include Visual Studio, Eclipse, REPL.it, and command prompt. | Frequently employed programming languages include Python, Java, CSS, JavaScript, HTML, PHP, Ruby on Rails, C++, and more. Developers utilize tools such as REPL.it, Node.js, GitHub, Visual Studio, and command prompt (PowerShell) to enhance their development workflow. |

## 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**: To ensure a smooth start and wide software availability, I recommend launching The Gaming Room on Windows devices. Windows, being the most widely used operating system, offers ample development options and a robust support community. By utilizing Windows, we can engage with a larger customer base due to its familiarity. It's important to note that mobile games and computer games differ in various aspects, such as usage, user experience, gameplay, and design. Game developers on all platforms, including Windows, strive to create high-quality games to compete effectively. Hence, Windows is highly recommended for The Gaming Room.
2. **Operating Systems Architectures**: When it comes to mobile gaming, the Arm architecture shows promising potential despite its lower power compared to x86. On the other hand, x86, which includes Intel and AMD processors, offers substantial power, a wide range of development options like Unity and Unreal, and exceptional hardware compatibility. Windows NT, the fundamental structure of Windows Operating Systems, operates on a hybrid platform with two layers. The user layer grants access to essential resources for running applications and interacting with input/output operations, while the kernel layer enjoys complete hardware access and enables services for graphical interfaces, system resources, and other functionalities.
3. **Storage Management**: The best option for storage is to go with an SSD instead of an HDD, as SSDs offer superior speed and are more modern, even on new consoles. When it comes to SSDs, there are different options available, such as SATA and NVMe. For those looking for a system that is ready for the future, considering PCIe 4 is recommended. To strike a balance between price and speed, a Q Cell NVMe SSD is a good choice. Additionally, for users with limited storage capacities, utilizing Microsoft Account OneDrive is a great solution. It provides 5GB of personal cloud storage, and Office 365 subscribers get access to 1TB of OneDrive storage. Windows 10's Storage Sense feature further enhances storage management by allowing easy scrutiny of files, choosing save locations for apps, and utilizing cloud storage. The built-in storage system in Windows 10 ensures convenient file creation and organization for large projects, minimizing the risk of loss or accidental deletion.
4. **Memory Management**: To store a large picture database or library for your game project, it's recommended to allocate memory outside of the default picture folder. This ensures the project stays secure and organized on your computer, even when working with your IDE and opening files. Windows utilizes a feature called Windows Memory Management, which reserves a portion of storage space for virtual RAM. Processes not actively in use are moved to this virtual memory, protecting the live processes in physical RAM. In 32-bit Windows, each process has a 4-gigabyte virtual address space, while in 64-bit Windows, it's 8 terabytes. Threads within a process can't access memory of other processes, safeguarding against corruption.
5. **Distributed Systems and Networks** Network games often require a shared database for players interacting over the network. The Artery architecture offers a solution by providing a high-level application program interface and improving network performance. Although a console release is not planned, distributing "Draw It or Lost It" on the Xbox app for Windows would be advantageous. Using Develop 4, a cross-platform IDE, allows for game creation and export to various platforms for cross-play. Ensuring strong servers and backup power is essential to support a large player volume and prevent outages.
6. **Security**: Windows includes built-in security protection software, but it is advisable to use additional security measures to protect user data. Windows Defender, a standard antivirus software, is pre-installed on Windows devices. Real-time scanning and automatic updates help keep the system and user information secure. However, it is important to be cautious when transmitting sensitive data during gaming. Taking data security into account during game development can provide competitive advantages, including compliance with regulations like GDPR.