

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc170035200)

[Table of Contents 2](#_Toc170035201)

[Document Revision History 2](#_Toc170035202)

[Executive Summary 3](#_Toc170035203)

[Requirements 3](#_Toc170035204)

[Design Constraints 3](#_Toc170035205)

[System Architecture View 3](#_Toc170035206)

[Domain Model 3](#_Toc170035207)

[Evaluation 4](#_Toc170035208)

[Recommendations 6](#_Toc170035209)

[References 8](#_Toc170035210)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/23/24 | Westley Hunter | Completed Executive Summary, Design Constraints, and Domain Model sections |
| 1.1 | 6/5/2024 | Westley Hunter | Completed Evaluation Section |
| 2.0 | 6/23/2024 | Westley Hunter | Completed Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room seeks a web-based game that can serve multiple platforms based on their Android app called ***Draw It or Lose It***. The Gaming Room has outlined specified technical requirements:

* Games can have one or more teams involved
* Teams can have multiple players
* Game and team names have no duplicates
* Player names have no duplicates within a team
* Only one instance of a game can exist in memory at a time

The Gaming Room has outlined the below business requirements:

* Ensuring the development team has web-based and cross-platform experience

## 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)

1. Team member with cross-platform experience
   1. **Implication:** Without this, time will have to be allocated for education in cross-platform development and completion of objectives will be significantly delayed or absent
2. Only one instance of game can exist in memory at a time
   1. **Implication**: Create code to limit instances and code to test there is only one instance
3. Game and Team names have no duplicates
   1. **Implication**: Constructing safeguards to prevent duplication
4. Methods to add teams to games and players to teams
   1. **Implication:** Employing encapsulation and abstraction to streamline execution and reduce redundant code

## [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)

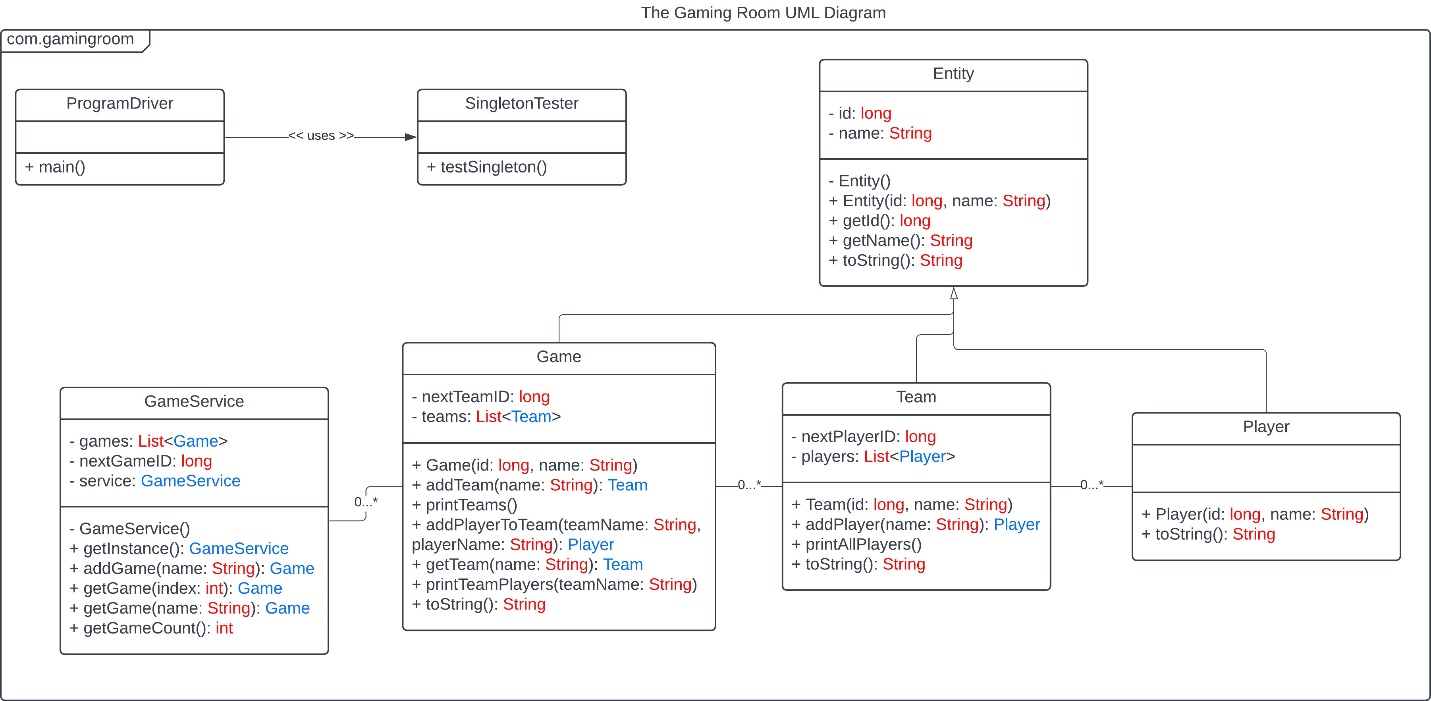
Below is an updated UML class diagram representative of the project code that I will be submitting. In the top left corner, you can see that ProgramDriver has only one public method and that is main where all the code will be run. Additionally, SingletonTester has only one public method. The ProgramDriver class uses the SingletonTester as a way to ensure that only one instance of the game exists in memory at a time.

Furthermore, we can see that the classes of Game, Team, and Player all inherit from the base Entity class. This means that each of the subclasses have all the same variables and methods listed in the Entity class diagram.

When examining Player, we can see its’ constructor and the overridden toString method which is a great example of polymorphism.

When looking at Team, we see the private variables of nextPlayerID and the players list of the Player object. nextPlayerID helps ensure that each player has their own unique ID. The players list allows for seamless searching of all existing players in a team. The added addPlayer method gives the functionality of adding new players to a team and the printAllPlayers method was implemented for debugging purposes but also serves to show all players on a team.

When moving to the Game class, we can see the variables of nextTeamID and the list of teams. nextTeamID ensures that each team has a unique ID and the teams list provides an easy way to look up all teams in a game. Furthermore, the additional methods of printTeams, addPlayerToTeam, getTeam, and printTeamPlayers serve to ensure the codes correct functionality and debug any issues. However, they also abstract the underlying processes for a working program. For example, the printTeamPlayers method uses the getTeam method to prevent redundant code as well as the printAllPlayers method in the Team class which prevents the Game class from accessing the players list in the Team class. Additionally, there are 0 to many instances between GameService and Game, between Game and Team, and between Team and Player.

****

## [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.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Pros:  Easy administration, great support, easy workload distribution  Cons:  Only runs on Apple hardware, which is pricey, not many 3rd party applications | Pros:  Open-source, high security, high-level language compilers  Cons:  Lack of long-term support for distributions and complex operations, difficult to use | Pros:  Intuitive GUI, 3rd party application support, many versions to choose from  Cons:  User-based licensing, more security threats | Pros:  Cheaper  Cons:  Does not have steady and stable internet connection, not a lot of computing power, may even be against your ISP terms of service |
| **Client Side** | Expensive to acquire Apple Hardware with minimal skills needed to navigate the OS and intuitive GUI | Low in terms of cost but high level of experience or training needed to operate the OS | Easy to understand and operating without a deep need for experience. Slightly expensive | Expertise in specific mobile OS and not a suitable environment |
| **Development Tools** | Languages would be Swift, HTML, CSS, and JavaScript. IDEs would include Visual Studio, Xcode, and Github. Additionally, the MacOS Server app is required. | Languages would be C, Python, HTML, CSS, and JavaScript. IDEs would include Visual Studio, PyCharm and Github. | Languages would be Python, HTML, CSS, and JavaScript. IDEs would include Visual Studio, PyCharm and Github. | Languages would be Python, Swift, Java HTML, CSS, and JavaScript. IDEs would include Visual Studio, Xcode, IntelliJ, PyCharm and Github. |

## 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 best operating platform for the Draw It or Lose It program would be Windows. Windows is an intuitive software with seamlessly integrating with a significant amount of 3rd party applications. Furthermore, selecting a specific version that best suits the program is easy. While licensing cost scale with the number of users, the current infrastructure and Windows ease of use make it the best option.
2. **Operating Systems Architectures**: Windows is a good choice for this piece of the development plan for the Draw It or Lose It program. The Windows kernel is robust and has a leg up on the competition in terms of security, I/O, and device driver models (Cortex, n.d.). Furthermore, Windows has a wide variety of applications for use on their platform that can be leveraged to achieve a lot of goals.
3. **Storage Management**: When talking about storage management, Windows is a great option as there are built in features to automatically delete files, identify unused files, and even relocate storage to other devices (Microsoft, n.d.). By using Windows Storage Sense, the Windows operating system will automatically delete unnecessary files and free up space without the user spending any time deleting files. However, if storage becomes scarce, the user can receive cleanup recommendations from the system on temporary and unused files to delete. Lastly, if any more storage cannot be found on the device, then Windows allows for a seamless transfer to a removable storage device.
4. **Memory Management**: Within Windows, each process has a virtual address space with a capacity of up to 4 gigabytes of memory of space (Microsoft, 2021). Therefore, the Draw It or Lose It program will have no problem storing four photos at a time in the memory process and could even store more to reduce the amount of loading screens.
5. **Distributed Systems and Networks**: The easiest way to ensure that the program runs on all Operating platforms is to develop the project in the Java programming language. Java has a unique feature that all final and runnable programs are incorporated with a Java Virtual Machine (JVM). When the program is executed, the JVM boots up a virtual machine that allows the program to run anywhere. The JVM capability will guarantee that the Draw It or Lose It program will work on Windows, Linux, Mac, and mobile platforms as well. In order to mitigate outages, the server side should be equipped to switch to new servers when an outage is detected. Furthermore, the server side should monitor the amount of client requests and scale the server usage accordingly to prevent any outages.
6. **Security**: To ensure that the Draw It or Lose It program maintains a tight security stance, development will use common and tested programming and design patterns. This will ensure that Object-Oriented Programming principles are followed and will include patterns like singleton and iterator. Furthermore, both the client and server side will properly protect any sensitive information that is being transferred by ensuring that it is encrypted up to modern standards. Additionally, Windows has an included program called Windows Defender Antivirus that scans for malware and actively protects the servers. To protect against memory corruption, Windows has Control Flow Guard that combats memory corruption vulnerabilities and ransomware attacks, restricts capabilities of server to whatever is currently needed, and marks it harder to exploit arbitrary code (Microsoft, 2022).

## References

Cortex, B. (n.d.). *How does the Windows Kernel work compared to the Linux Kernel?* Retrieved from Quora: https://www.quora.com/How-does-the-Windows-Kernel-work-compared-to-the-Linux-Kernel#:~:text=The%20NT%20(Windows)%20and%20Linux,O%2C%20and%20device%20driver%20models.

Microsoft. (2021, January 7). *About Memory Management*. Retrieved from Microsoft: https://learn.microsoft.com/en-us/windows/win32/memory/about-memory-management

Microsoft. (2022, February 8). *Control Flow Guard for platform security*. Retrieved from Mircosoft: https://learn.microsoft.com/en-us/windows/win32/secbp/control-flow-guard

Microsoft. (n.d.). *Free up drive space in Windows*. Retrieved from Microsoft: https://support.microsoft.com/en-us/windows/free-up-drive-space-in-windows-a18fae02-a0fa-8df9-9838-8970f9939de4