



GamingRoom  
**CS 230 Project Software Design Template**  
Version 1.0

## Table of Contents

<b>CS 230 Project Software Design Template</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>Document Revision History</b>	<b>2</b>
<b>Executive Summary</b>	<b>3</b>
<b>Design Constraints</b>	<b>3</b>
<b>System Architecture View</b>	<b>3</b>
<b>Domain Model</b>	<b>3</b>
<b>Evaluation</b>	<b>4</b>
<b>Recommendations</b>	<b>8</b>

## Document Revision History

Version	Date	Author	Comments
1.0	09/15/22	Cody Faircloth	<ul style="list-style-type: none"><li>• Implementation of Singleton GamingService class</li><li>• Implementation of inheritance from Entity class to Game, Team, and Player classes</li><li>• Completion of ProgramDriver for testing purposes</li></ul>

## **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**

The Gaming Room would like an environment for their new game service Draw It or Lose It that saves data for games being played, teams in each game, and players on each team. Each instance of either a game, team, or player should have a unique name and a unique identifier.

## **Design Constraints**

The application should have a single instance of GamingService so only one object can create games. Multiple games should be able to be created but each game should have a unique name and identifier. Each game should have multiple teams assigned to the game. Each team should also have a unique name and identifier. Each team should have multiple players assigned to the team. Each player should have a unique name and identifier. Only one instance of each specific object can exist at a time. For example, two games, teams, or players cannot have the same identifier or the same name.

## **System Architecture View**

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**

The GameService class is a Singleton class meaning one instance of a GameService object can be created by the ProgramDriver. The ProgramDriver uses the SingletonTester to ensure only one instance of GameService exists at any point. Multiplicity is present between GameService and Game, Game and Team, and Team and Player. This means that for each instance of GameService there will be zero to many instances of Game, for each instance of Game there will be zero to many instances of Team, and for each instance of Team there will be zero to many instances of Player. The Game, Team, and Player Class all inherit from the Entity class meaning all child classes share variables, constructors, and methods of the super class or parent class.



### Evaluation

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
--------------------------	-----	-------	---------	----------------

<b>Server Side</b>	<p>MacOS is an expensive brand to use since Apple products are required to run MacOS. However, MacOS provides excellent application abilities. MacOS provides web hosting abilities along with flexible terminal commands. MacOS is ran on hardware that is usually very high quality, while expensive, with regular upgrades to the hardware. It also provides many web browser options. However, Mac OS is not the preferred operating system to use solely for web browsing.</p>	<p>Linux provides features similar to Mac OS with a lower cost to the company. Linux is generally considered to be a preferred operating system with great security provisions. Development and deployment would also be very swift. One disadvantage of this operating system is the difficulty of application storage management.</p>	<p>Windows includes the largest amount of software choices of all operating systems. Windows is the predominant operating system and is widely used. To most users Windows has a familiarity. Windows has consistent updates. The main disadvantage of Windows is its comparatively weak security provisions and its susceptibility to viruses.</p>	<p>Mobile Device operating systems can manage calls to the server side of the application to optimize queries on databases. Mobile Device operating system does not allow for the accesses or modification of code from the user. Mobile Device operating systems allow for great portability along with greater compatibility at a better price than other operating systems. There are security issues with this type of operating system. This operating system also would require cloud services or physical services integrations for the server side.</p>
--------------------	---	---	---	---

<b>Client Side</b>	<p>MacOS provides a wide range of supported web browsers which includes development tools. These development tools easily allow for cross-browser testing for the client. The time for development and deployment using MacOS would be a short to medium duration in terms of time. The cost will be comparable to Windows, however the level of expertise needed would be relatively low.</p>	<p>Linux would have a similar development time as MacOS and Windows; however, more expertise would be needed. The overall cost would be significantly less than MacOS or Windows. Linux includes tree file accommodations and works with every web browser because of the availability of open-source software.</p>	<p>Windows has a slightly lower cost than MacOS but a higher cost than Linux. It is less user friendly than MacOS but more user friendly than Linux. Time required for development and deployment would be similar to MacOS and Linux with more expertise needed than MacOS but less expertise needed than Linux.</p>	<p>This operating system provides more flexibility and simplicity to the end user or developer. This is the option with the best cost. More expertise is required as mobile operating systems are “quirky” but this company already has experience with the Andriod operating system. This operating system would also take more time to developm</p>
--------------------	--	---	---	---

<b>Development Tools</b>	<p>MacOS supports HTML, CSS, and JavaScript with multiple libraries to support the web portion of the applications. MacOS supports C++, C#, Java, and Python among other languages to support the application portion of the program. Applications must be approved and reviewed by Apple. Additionally, the Apple Developer Program costs \$99 per year. Apple supports a plethora of IDEs.</p>	<p>Linux supports HTML, CSS, and JavaScript with multiple libraries to support the web portion of the applications. Linux supports C++, C#, Java, and Python among other languages to support the application portion of the program. Linux includes no licensing costs. Linux supports a plethora of IDEs.</p>	<p>Windows supports HTML, CSS, and JavaScript with multiple libraries to support the web portion of applications. Windows supports C++, C#, Java, and Python among other languages to support the application portion of the program. There is no licensing costs for Windows. Windows supports a plethora of IDEs.</p>	<p>There are several different operating systems for mobile devices each of which supports different programming languages. There are licensing costs for iOS. SwiftUI is a major IDE for mobile operating systems.</p>
--------------------------	--	---	---	---

## **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 Windows for the expansion of Draw it or Lose it. Windows seamlessly integrates with Android builds. You can simply download Android Studio to begin development on the Android operating system. There are also many other cross-platform IDEs to choose from such as Xamarin, React, or Cordova. Windows allows for application to be easily developed and deployed for all Windows PCs and systems running Windows. Windows also allows for a variety of emulators and virtual machines in order to test development on other operating systems. Windows is the most used operating system by businesses and the public. Additionally, Windows accounts for the largest user base in terms of gaming.

### **2. Operating Systems Architectures:**

Windows architecture works by separating the operating system into two unique modes with specific roles and functions. These modes are user mode and kernel mode. User mode functions include most of what the user interacts with. This includes User Interfaces, User Experiences, and the like. This mode deals with everything on the front end of the operating system. Kernel mode functions include dealing with input and output devices, memory and storage management, networking such as connecting to Wi-Fi and other devices, hardware management, and running system routines such as boot order. Windows uses a user-friendly directory structure to handle the storage of data. This directory structure has changed little with recent iterations of Windows allowing for a familiar experience regardless of which version of Windows is used. Windows systems allow for modularity of hardware allowing for a large variety of builds depending on user needs. For example, if you need more processing power, you can switch to a more powerful CPU without switching operating systems. This says more about the machine itself rather than the operating system; however, other operating systems, such as Mac OS, do not allow this level of modularity.

### **3. Storage Management:**

Storage management will be an important aspect of this application. I recommend the implementation of Microsoft Azure. Azure is Microsoft's cloud computing environment. They provide competitive pricing, excellent user support, and many notable features, along with continuous updates and improvements. Since this application will have much data to store, increasing as game popularity increases a cloud-based storage solution will be the best option. Adding or removing memory is simple with Azure and allows for the game to scale over time without expending unnecessary money. The library of 200 photos would be stored on Azure File Share which starts at 1.6 GB per user but can be increased as the library grows. Additionally, Azure allows access to Azure SQL Database which is a cloud storage database. This would be a perfect solution for the storage of user data, such as username, password, and other account information.

### **4. Memory Management:**

Windows 11 is the latest version of Windows. Windows 11 uses more RAM than Windows 10 due to the number of added features. However, this increase in RAM usage does not equate to worse memory management or worse performance than Windows 10. Depending on the amount of RAM installed into your machine, the operating system automatically allocates more or fewer resources to different



applications depending on performance needs. The efficiency of loading memory is primarily implemented using disc paging and demand paging. These two processes act as an extension of the system's physical memory or RAM. Disk paging accomplishes this by reserving a portion of the hard drive as additional RAM. Demand paging separates larger tasks into smaller processes and only loading those processes into memory when required. Windows also allows for the use of entirely virtual memory address space. Given the requirements of this application, the utilization of virtual memory would be the perfect solution.

#### **5. Distributed Systems and Networks:**

Azure provides ease of access when working with distributed systems and networks. Azure boasts maximum uptime along with cloud-based email alerts in the occurrence of an outage. Azure features also include application insights, logging, and monitoring solutions for the server. These features will be essential as the application grows over time. As more games occur and more players are in said games, it will be critical for outages and issues with the cloud to be caught and corrected immediately. Outsourcing the network load to Azure allows for another team to focus on this aspect while the development team can focus on improving the application for the user.

#### **6. Security:**

Along with storage management and distributed network solutions, Azure also provides a solution for security. The application will secure user data and information by having the devices connect to the Azure App Service running on an App Service Plan. This will then implement the Azure Active Directory to log users into their accounts. Azure offers security features such as IP configurations to whitelist access to sensitive data such as a player's personal information. Azure also allows for the utilization of storage within a VPN to further enhance the security of cloud data. There is also an option to require IP whitelist access, a valid password, and SSL connectivity to access the database of user data which further secures user data. Additionally, as a final safeguard, Azure offers the option to encrypt user data more thoroughly in the event of a breach. The only downside of this solution is the requirement for an active internet connection in order to communicate with the cloud. However, from the parameters of the application, an active internet connection will be required to run the game anyway.