

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 | 05/20/21 | Brittany-Winters | Includes executive summary, design constraints, domain model, evaluation, and recommendations. |
| 1.1 | 06/06/21 | Brittany-Winters | Updated server, client, and development tools. |
| 1.2 | 06/18/21 | Brittany-Winters | Updated recommendations. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room is looking to develop a web-based project based on their Android mobile game “Draw It or Lose It”. The client requires direction in setting up an environment that will allow players to access their game on multiple platforms. Ultimately, this will expand their customer base and promote growth within the company.

## [Design Constraints](#_2et92p0)

1. **Platform Constraint:** Creating a web-based version of a game that already exists on the Android platform could be a challenge since we will need to adapt to new or different platforms. We can accomplish this by ensuring that our team is well rounded and understands the differences in each platform (Windows, MacOS, Linux) for proper implementation and use.
2. **Time Constraint:**  Since we are developing “Draw It or Lose It” to be played on multiple platforms, time can become a constraint. Unexpected problems with the development for one platform could affect the development of others, overall delaying the project. We can attempt to overcome this challenge by delivering very small increments at one time. Small batch deliveries can help us keep track of what is working well and what we can do better in order to quickly make changes if needed and stay on track with the schedule.
3. **Branding and Marketing Constraint:** Since we are developing a game based on an already existing game, branding and marketing can be a constraint. Users of the existing game may be hesitant to play on a completely new platform. In order to make users comfortable, we should stick closely to the already successful game layout and branding plan.

## [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 Program Driver has one public main function and its operation is to start the program. The Program Driver is used to drive the multi-class program. The Program Driver <<uses>> the Singleton Tester Class. <<uses>> refers to the dependency relationship since The Program Driver relies on the operation of the Singleton Tester. The shaded arrow also represents association.

The Singleton Tester class has one public main test function and it can be accessed by the Program Driver. The Singleton Tester contains a Singleton design pattern.

The Game Service class has five private attributes, one private operation, and seven public operations.

The Game class has one private attribute that includes a list. The Team class holds a private list and three public operations. Finally the player class two public operations.

0…\* between classes represent a zero to many relationship, while the hollow arrow leading into the Entity class represents inheritance. The UML shows hierarchical inheritance, where the Entity class is the base class for the Game, Team, and Player classes. We practice encapsulation by making certain attributes and operations private and only assessable by designated classes. Encapsulation makes are code reusable and protects our data from manipulation. We also use method overloading by having multiple operations with the same name, but with different parameters such as in the Game Service class with the getGame() function.

****

## 

## 

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | One of the greatest advantages of Mac is its integration with other iOS devices. Because less people use MacOS there are less security issues. A disadvantage to MacOS is that it offers less customization.  MacOS does permit the use of servers provided by Apple. Although hosting through MacOS is not popular, It is inexpensive to get started in comparison to Windows if you choose to do so. | One advantage of using Linux for hosting is that it is open source. Linux is also speed and stability which will help applications run smoothly. A disadvantage is that Linux is not fully compatible with Microsoft products and technologies.  As mentioned, Linux is open-sourced which makes it a popular low cost choice for hosting. There may be a learning curve with Linux, as most people are mainly familiar with Mac and Windows. | An advantage of Windows is that it is fully compatible with Microsoft technology. Windows can support SQL databases, and Visual Basic Scripts. Windows is also slower compared to Linux.  Using a server based on Windows can be expensive to start. However, the quality and reliability makes this a good choice for those looking for stability. | Mobile devices are typically much faster when it comes to running applications. A disadvantage is that multiple versions need to be developed in order to use on different mobile platforms.  Hosting on a mobile device can cause issues since mobile devices lack some of the capabilities that are present on traditional computers. Mobile device hosting is the most inexpensive option. |
| **Client Side** | Accessibility is a disadvantage when developing software for MacOS. Mac is less used overall in comparison to Windows. The cost of hosting here is also more expensive compared to both Linux and Windows.  The cost of using a server through MacOS can be expensive in comparison to Linux, and Mobile Devices. | Since Linux is open source, this means hosting will be inexpensive compared to other options. A lot developers prefer Linux because of its stable, fast and secure features.  Since Linux is not the most popular choice to use, maximum time is required to support the system. However, Linux’s open source feature can make the extra time worth it. | A disadvantage to using Windows to host is that the fees can be expensive. Windows is a popular product that most people are familiar with. Finding experienced Windows developers should be an easy process.  Hosing through Windows can be less time consuming, many people are familiar with Windows. This can become costly as Windows is not open sourced. | The cost of using Mobile Devices can be quite high. This is because mobile applications require constant maintenance, and updates. Accessibility is an advantage when developing a mobile app because most people have a mobile device.  Cost and user experience are advantages for mobile devices. However, hosting on a mobile device will require a lot of maintenance |
| **Development Tools** | MacOS uses Objective-C most commonly as a programming language. Perl, C++, HTML and Java are also often used. Xcode is a popular IDE for MacOS along with Eclipse.  MacOS also uses the programming language Swift for its applications. Swift can be used on Xcode, Atom, and Apple Developer Tools. There are free and paid versions of these IDEs. | Some programming languages that Linux applications use include C, HTML, Java, and Python. Common IDEs for Linux include Bluefish Editor, Geany, Gedit, and Atom. There are free and paid versions of these IDEs. | Many Windows applications use C# which was developed by Microsoft this purpose. Popular IDEs for Windows applications are Microsoft Visual Studio, and Eclipse.  These IDEs have both free and paid versions. | Mobile Applications can be developed using many different languages such as Swift, Kolton, HTML, or Java for example. Some IDEs include Android Studio, Visual Studio, or Xcode. These IDEs have both free and paid versions. |

## Recommendations

1. **Operating Platform**: The Operating Platform I would recommend in order to successfully expand Draw It or Lose it to other platforms would be Windows. I suggest Windows because the game is already developed for Android. This means we could make a fast transition using some of the same familiar tools. Furthermore, Windows is an extremely popular operating system, insuring a wide range of accessibility. The wide availability also ensures that developers will have many options for IDEs.
2. **Operating Systems Architectures**: Microsoft is responsible for producing the Windows operating systems. The architecture has both a user and kernel mode. The design of the Windows Operating System allows us to use kernel processes without directly affecting them. This means that the processes affiliated with the program will not affect the processes of the Operating System. I/O requests are processed by using I/O request packets.
3. **Storage Management**: Windows is also a popular option because of its Storage Management options. The Windows Operating Systems has features such as the disk management that can be used to facilitate advanced management. Users have access to the Disk Clean Up feature which allows storage management on a smaller scale.
4. **Memory Management**: Paging is used in most operating systems today. Paging involves breaking physical memory into fixed blocks that are known as frames while breaking logical memory into equally sized blocks called pages. The pages are then loaded into available memory when a process is ready to be executed. Ultimately, paging allows programmers to discontinue use of contiguous allocation of physical memory.
5. **Distributed Systems and Networks**: We use distributed systems to accomplish problems across multiple computers. We can use the client-server distributing system in order to achieve a reliable application for users. Our central server will provide the needed application data so that other systems/clients are able to use resources as needed in relation to the game and its function.
6. **Security**: Windows has many security features such as Windows Defender Antivirus and Windows Sandbox. Windows also promotes regular system updates that help guard against cyberattacks. Passwords, user accounts, and data encryption are also used to promote security.