

<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 | <07/12/22> | <Michael Straughter> | Made changes to the Executive Summary, Design Restraints, System Agriculture Review, Evaluation, and Recommendations. |

**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)

Our newly required client, The Gaming Room wants to develop a web-based game called Draw It or Lose It. Serving on multiple platforms, Draw It or Lose It will render images from a large library of stock drawings as clues. This app will be only available on Android only. To help facilitate the development of the web-based version of this gaming app, software applications will be created using Java. Using design patterns that ensures a single class can only have one object.

## [Design Constraints](#_2et92p0)

* No specific time limit was given by the costumer
* The platform the game is being developed for is specifically Android/Linux-based
* Technical bugs and limitations

## [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)

Every object in the Game Service class is associated with zero or more objects in the Game class, Team class, and Player class. The Entity class inherits from the three classes Game, Team, and Player allowing the reuse of the existing code from all three classes. The fields and methods are all inherited from each class. By making constructors private in the Game Service, object creation is restricted outside of that class.

**"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** | The Mac is known for having an instinctive design that allows you to work easily between different windows, tabs, and applications all at the same time. The functionality has many popular commands that come in handy for the use of software development. Some weaknesses of the Mac would be the limited storage options, and inadequate computing proficiencies that may be experienced for those who develop more advanced web games. | Linux is designed to support high volume and multithreading applications such as web servers, database servers, and file servers to name a few. Advantages of Linux are that it is an open-source operating system. This means that its source code is available for anyone. Some disadvantages of Linux are the limited applications that work for other operating systems that are not compatible with Linux. Also, user experience has reported as being different, and at sometimes timely to learn. | The characteristics of Windows are its multitasking and multithreading capabilities. With its virtual memory management system and symmetric multiprocessor, it is ideal for an optimal experience on the web. A major advantage of Windows is that it just seems to be easier to use. When you add that factor in with the compatibility and functionality of web-browsers you can understand the popularity of this OS. Some disadvantages of Windows would be the poor security which allows user to be more susceptible to hacking as opposed to other operating systems | The size and, touch screen interface are major characteristics of most every mobile device. Advantages of Mobile Devices are the flexibility of service that can be offered for almost any cellphone. With access to apps and networking capabilities its ideal for various platforms of communication. Disadvantages can be seen as the limited dimensions of the screen, and the vulnerability to SPAM. |
| **Client Side** | Here is what to consider when supporting multiple types of clients pertaining to Mac. The average cost for web development is around $60,000 - $300,00. The time that that estimated cost comes to is about 1,200h. Expertise for web development depends on the complexity level. The number of features implemented in the product determines the complexity. | Here is what to consider when supporting multiple types of clients pertaining to Linux. An average Software development for a medium sized app would cost $70,000 to $100,00. Coming to about 4 -6 months or 2000hrs. Again, most expertise will depend on the complex of the software and the level of customization. | Here is what to consider when supporting multiple types of clients pertaining to Windows. Windows is capable of handling various task for developers. Windows supports many programs and languages. The security features will help keep the clients shielded from all types of threats. The OS is also easy to upgrade which will be very useful when considering price and time. | Here is what to consider when supporting multiple types of clients pertaining to Mobile Devices. Mobile development varies in different parts of the world and so will the cost. Cost generally reflects off the economic development in a specific country. For example, the average developer makes $120,00/ year in North America. This is a big gap for a developer in countries like India, where the average developer makes about $4,100/year. Another cost would be the maintenance components which is usually 15 – 20% of the original development cost. |
| **Development Tools** | Relevant programming languages and tools that are used to build this type of software for deploying on Mac consist of IDEs like Visual Studio Code and PhpStorm. You can write in C#, Javascript, and many other languages using these IDEs. Syntax highlights, compiling and debugging are all useful tools for building the software. | Relevant programming languages and tools that are used to build this type of software for deploying on Linux are Komodo, and Sublime text to name a few. Supporting various languages like Python, C, and JavaScript. Customizable files are a valuable tool that go into software development. | Relevant programming languages and tools that are used to build this type of software for deploying on Windows is again Visual Studio Code. Visual Studios code is developed by Microsoft and can making coding faster. It is also equipped with a community of support. Tools that also go into building a successful software deployment through Windows is enabled auto complete and great in-built templates. | Relevant programming languages and tools that are used to build this type of software for deploying on Mobile Devices are Android Studio and Xcode to name a couple. Tools that go into helping the software development are GUI tools integration, and Faster coding and speedy iteration. |

## 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**: An appropriate operating platform that will allow the client to expand to other computing environments would be the use of Codename One. Codename One is an open-source cross-platform aiming to provide code for various mobile and desktop operating systems.
2. **Operating Systems Architectures**: The core of the Linux operating system is made up of the Kernel, System Library, Hardware layer, System, and Shell utility. These sections combined make up the software within a computer system.
3. **Storage Management**: An appropriate storage management system that can be use with the operating platform is Seafile. Seafile provides desktop clients for Windows, Linux, and mobile clients for Android. Useful features of Seafile are file editing, differential sync to minimize the band width required, and client-side encryption to secure client data.
4. **Memory Management**:

Linux uses memory management techniques for the Draw It or Lose It software by containing the amount of user space programs and kernel internal structures. The Linux subsystem memory management is responsible for managing the memory inside the system.

1. **Distributed Systems and Networks**:

Draw It or Lose it will be able to communicate between various platforms using shared memory. The usage of familiar memory read/write operations, shared memory bandwidth, and low overhead of ordinary memory read/write operations.

1. **Security**:

User information on and between various platforms will be protected by preventing any area of the kernel with executable memory to be writable. This includes kernel text, kernel modules, memory etc.