

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 | 07/12/2022 | KaLee Li | 1. Including information for executive summary and Domain model in this document. 2. Continue developing the Java Application |
|  | 07/29/2022 | KaLee Li | 1. Evaluation |
|  | 8/11/2022 | KaLee Li | Analyze the characteristics of and techniques specific to various systems architectures and make recommendation for client |

[Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants to develop a web-based game that serves multiple platforms based on their current game, Draw it or Lose it, which is currently available in an Android app only. In this document, we will address some of the requirements and possible design constraints that may arise during the process of development.

Requirements for Draw it or Lose it:

1. More than one team involved
2. Each team has multiple players
3. Game and team names must be unique to allow users to check the weather a name is in use when choosing a team name
4. Only one instance of the game can exist in memory.
5. The application will render images from a large library of stock drawings as clues.
6. Four rounds of play lasting one minute each
7. Drawing are rendered at the 30-second mark
8. If the team does not guess the puzzle before time expires, the remaining team has an opportunity to offer one guess with 15 second time limit.
9. This game will run on multiple platforms.

Business Constraints:

1. Schedule and Budget

* To work with a fixed budget, this software development will require an extended schedule because we will need a lesser amount of team members to work on different areas of this project
* Increase budget to hire more developers to get the project completed as scheduled

1. Team Composition and make-up

* Requires a specific to have knowledge of web-based app development. Otherwise may need to provide training

1. Software Licensing restrictions or requirements

* No licensing is necessary for creating a game. However, it is a good idea to have one. Therefore will need to do more research on licensing and publishing. It May or may not require extra cost.

1. 3D modeling, sound for software

* Since this is a drawing and guessing game, it would be a good idea to have 3D modeling and sound. Will the gaming room provide an in-house graphic designer or hire someone?

## [Design Constraints](#_2et92p0)

Technical Constraints:

1. Program Language

* Java

1. Hardware requirement

* The minimum necessary to run this software smoothly.
  + - * CPU is powerful enough to handle the execution, speed, and perform multiple tasks simultaneously.
  + Memory (RAM)
  + I/O
  + Sound card & graphics cards

1. Operating system

* Multiple platforms

1. Cloud Environment

* Limitation of data used
* Cost of after limit is reached if any

## [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 UML class diagram below visualizes the way how the game system is designed. It shows the attribution (variables, data containing value), methods (a function that allow us to specify any behavioral feature of a class), relationships, and multiplicity of each class.

The Entity class is the superclass (parents) and have several subclasses (children). The subclasses are Game, Team, and Player. They shares the all the attributions and methods from the superclass, such as id, name, Entity(), Entity(id, name), getid(), getname(), toString(). Each subclasses has their own attributions and methods.

The subclasses have a associated relationship with each other. The GameService is associated with game. The game is associated with a team. The team is associated with player. 0…\* equal to 0 to many meanings that it can have none or many games, teams, and players.

The programDriver and Singleton Tester have a direct association between the classes. The arrow that indicated direct association is from ProgramDriver to SingletonTester with <<uses>> signifies that Programdriver uses SingletonTester to check validation for this example.

**The Primary Benefit of OOP**

* **Portability** – Also referred to as modularity. OOP, we program in a series of classes. This example has Entity, Game, Team, Player, GameService, ProgramDriver, and SingletonTester class. If a problem arises with one method, we can check the method of that particular class, which makes troubleshooting efficient. (Lavieri, 2019)
* **Inheritance** – Game, Team, and Player class are the subclasses that inherit from Entity Class, the superclass. The subclasses inherit all the attributions and methods from the superclass.
* **Encapsulation –** refers to the hidden nature of an object's data components. The data is there but cannot access directly because external entities cannot directly interact with that data (Lavieri, 2019). The attributions in all the classes were set as private (-)
* **Polymorphism –** In OOP, Polymorphism states that different types of objects can be accessed via a common interface. This can be achieved by writing overloaded constructors and methods. Polymorphism can also be achieved by having child classes override parent class methods. In this example, I do not see any overriding methods.

**"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** | macOS is designed by Apple. It can run several tasks simultaneously. Apple uses a combination of data centers and 3rd party cloud storage services. Although Apple claimed they do not offer to decrypt keys to the 3rd party cloud provider, I think the security is questionable. Mac OS server licensing is more costly with non-versatile upgrades and must be purchased for any add-on. | Linux is a versatile, Open source OS. Opensource and zero cost of entry, no charge for server licensing. Linux can run on a wide variety of hardware, from phones to supercomputers. Linux is not amateur programmer-friendly and not easy to understand. | Windows is one of the most popular OS. The majority of computers are running with window OS. Windows support various software and hardware. Windows server licensing cost is acceptable for small businesses. | Mobile OS is for mobile devices. Such as smartphones, tablets, smartwatches, etc. Android is based on the Linux kernel. iOS is based on Mac OS. Therefore, the pros and cons are the same |
| **Client Side** | Apple provides family-sharing features for the client.  Mac is easy to adjust and navigate.  Mac products are more costly compared to other alternatives. | Devices that run on Linux are more affordable, easy to use, and have a more interactive user interface.  Linux is not the best gaming, but it is coming together | Variety of pricing options for customers' needs. Window PC is the most popular, with better internal components and more liberty to configure their system—the best choice for gaming higher power GPU. The latest windows include antivirus protection. However, extra security and protection may be needed depending on user needs, which need to be purchased separately. | Mobile Device is portable and easy to carry for the client.  Many mobile device games are synced to the web-based games that the player is already playing. So, players can play a game that was saved anywhere.  Mobile devices are more affordable  Mobile Gaming is only for a short period because the battery drain is easier.  Mobile devices have smaller screens, so they are not as good for the eyes. |
| **Development Tools** | We can use Java or C++ for this game. C++ is compiled directly into machine code without an interpreter at runtime, so c++ performs better. However, C++ uses a pointer extensively. Misuse of pointers can lead to a memory leak.  Java's garbage collection performs automatic memory management.  Java program run in sandbox to ensure no malicious program from an untrusted resource.  Now, most IDE is compatible with Mac, Linux, and Windows. However, there are couples of IDE that are great for Mac. X-code is the built-in IDE for macOS, a complete developer toolset for creating an app for all Apple products.  Xcode brings user interface design, coding, testing, debugging, and submitting to the App Store into a unified workflow. It is a free download to use, but there is a fee to register as a developer to publish | Eclipse is one of the best IDE for Java. Eclipse is free for everyone and is a cross-platform IDE for macOS, Linux, and windows. Eclipse has the latest java with the necessary support, custom compiler, easy Git, and Apache Maven integration. | Visual Studio is an IDE that Microsoft develops for windows. VS is an open-source tool that is free for everyone and a cross-platform IDE for windows, Mac OS, and Linux. | As mentioned for macOS, Xcode can be used for Apple mobile devices. However, it would not work for android devices. Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. (Meet Android Studio |, n.d.) on top of IntelliJ's powerful code editor and developer tools.  It is essential to use cross-platform mobile applications to avoid recoding if developers are planning to create an application for both ios and android. Such as Xamarin, flutter, and sencha.etc |

## 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 stay within the gaming room budget, I recommend using Linux because Linux is a versatile, Open-source OS with zero entry cost and no charge for server licensing. Linux can run on various hardware, from phones to supercomputers.
2. **Operating Systems Architectures**: The Linux architecture is largely composed of elements such as the Kernel, System Library, Hardware layer, System, and Shell functions. The kernel is responsible for each of the main duties of the Linux OS. System Libraries are a set of OS-implemented functions and do not require code access on kernel modules. The hardware layer of Linux is made up of several peripheral devices such as a CPU, HDD, and RAM.
3. **Storage Management**: For Draw or lose it, we are supposed to render images from an extensive library of stock drawings as clues, 200 high-definition images file, and each one is approximately 8 megabytes in size. Therefore, the built-in layer of the Linux file system in Linux OS can organize and arrange the images in the disk. Disk drives are the major secondary storage; most are magnetic disks or tapes. Modern disk drives are structured as one-dimensional arrays of logical disk blocks – one disk blocks are 512 bytes in size.
4. **Memory Management**: A Linux file system is a structured collection of files on a disk drive or a partition. A partition is a segment of memory and contains some specific data. In Linux machine, there can be various partitions of the memory. Generally, every partition contains a file system. Linux file system is a built-in layer of Linux OS, and it handles the data of the storage to arrange files on the disk storage.
5. **Distributed Systems and Networks**:

Our distributed system must need a network that connects all components so it can communicate between various platforms. We can use all components using IP addresses so databases, objects, and files can share between the machines. For distributed systems architecture, we will use a Client-server for multiple players in a team to share their drawings, make guesses or send them back to the server for the remaining team.

1. **Security**: Security is a must-have for the client. I would recommend that this game only allow one account per player. Also, using user authentication protects our user information between various platforms.

**Reference:**

Lavieri, Edward D. *Hands-on Design Patterns with Java: Learn Design Patterns That Enable the Building of Large-Scale Software Architectures*. Packt Publishing, 2019.

GeeksforGeeks. (2022, February 20). *Which Operating System to Choose For Web Development?* https://www.geeksforgeeks.org/which-operating-system-to-choose-for-web-development/

Rossignol, J. (2021, June 29). *Apple Reportedly Storing Over 8 Million Terabytes of iCloud Data on Google Servers*. MacRumors. https://www.macrumors.com/2021/06/29/icloud-data-stored-on-google-cloud-increasing/

Roomi, M., & Roomi, M. (2022c, May 15). *5 Advantages and Disadvantages of Linux Operating System | Drawbacks & Benefits of Linux Operating System*. HitechWhizz - The Ultimate Tech Experience. https://www.hitechwhizz.com/2022/05/5-advantages-and-disadvantages-drawbacks-benefits-of-linux-operating-system.html

*Xcode - Support - Apple Developer*. (n.d.). Apple Developer. https://developer.apple.com/support/xcode/#:%7E:text=Xcode%20is%20a%20complete%20developer,Store%20into%20a%20unified%20workflow.

*Windows Server 2022 Licensing & Pricing*. (n.d.). Microsoft. <https://www.microsoft.com/en-us/windows-server/pricing>

*Meet Android Studio |*. (n.d.). Android Developers. <https://developer.android.com/studio/intro>

*Linux Architecture - Detailed Explanation*. (2022, June 10). InterviewBit. https://www.interviewbit.com/blog/linux-architecture/