

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 | 11/11/2022 | Kyle Conley | Expand Draw It or Lose It across multiple platforms |
|  | 12/11/2022 | Kyle Conley | Offer 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)

The Gaming Room is interested in developing a web-based game, “Draw It or Lose It”, that is functional on multiple platforms. This game is currently only available for Android’s and The Gaming Room hopes to increase its value by expanding its functionality across other platforms. This is guessing game that displays graphical clues throughout four rounds of play, each round lasting one minute. The clues are iteratively displayed within the first 30 seconds of the round, if the selected team does not solve the puzzle within one minute, then the remaining teams have 15 seconds to present one guess for their team.

## [Design Constraints](#_2et92p0)

Create a version of “Draw It or Lose It” that is functional on multiple platforms based off The Gaming Room’s Android version.

*Limiting factors for this include:*

* Must operate on Operating Platforms such as Mac, Linux, Windows, and Mobile Applications
* Game operates with one or two teams
* Each team requires at least two players
* Game and Team names must be unique
* Only one instance of each game can exist

*Technical Constraints:*

* The game is already functional for Android mobile apps but must be modified to operate on iPhones along with PC operating systems such as Window’s, Linux, and Mac.
* Include existing features and design from Android version

*Business Constraints:*

* Designing multi-platform applications may require the use of multiple developing teams and additional testing measures may also be needed.

## [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)

**As seen below, the Entity class acts as our parent or super class and includes member and member functions that are inherited to the Game, Team, and Player classes. These three subclasses of Entity demonstrate the inheritance principle of object-oriented programming by allowing the derivation of all properties and methods from the parent class to the three child classes. The Game Service class then references the Game class, then the Game class references the Team class, then the Team class references the Player class. These references indicate an association relationship in which zero to many restraints are implied. Next, the ProgramDriver and SingletonTester classes are the two classes needed to demonstrate a singleton design pattern. As demonstrated, the ProgramDriver class contains the main() method and is the “driver” of the entire program.**

**"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** | Mac offers open-source software with an abundance of features, simple and organized user interface, higher security, cloud storage services.  Disadvantages include incompatibility with large collection of applications developed in Windows. Mac Os as a host server isn’t feasible or efficient for cross-platform integration. | Free and open-source software that is easily customizable, high level security features, and can handle large user capacity without decreasing performance. Can upgrade server hosting options to include unlimited web space and HTTP/2 cryptographic data transmission.  Linux is less popular universally and less intuitive. | Windows is the most widely used platform. This benefits software compatibility and ease of development. Windows is a widely known platform but is also associated with higher licensing cost.  Windows is closed source, so platform transparency and adaptability are difficult, but security is considered stable. | Hosting web-based applications from a mobile phone platform would not be ideal. Lack of reliable power supply and inefficient memory/ storage capacity make PC operating systems the better choice. |
| **Client Side** | Developing in Mac requires specific knowledge of the platform. This specialization could be more costly and possibly more difficult to obtain. Mac has features helpful in streamlining development more efficient, but only in Mac OS. | Linux will require a higher level of expertise for efficient support. Android runs on Linux so creating a web-based game in Linux will likely be easier. Linux developers are in high demand and will likely cost more than Windows. | Windows will likely be more costly since it is closed sourced. The level of expertise is not as crucial as Mac or Linux and is likely to be more efficient. | While mobile application developers are in high demand, they are also high sought-after careers. Cost is similar to Mac or Linux. |
| **Development Tools** | Mac uses Objective-C, C, and C++ for its core. Swift and Python can are also popular for apps.  Xcode is a popular SDK designed exclusively for Mac OS. Visual Studio and Eclipse are other good options.  Mac OS developer membership cost $99 annually and can be upgraded to include additional Apple analytical and testing services.  Additional teams will be needed to integrate across other operating platforms. | Linux uses mostly C programming languages. Linux is known to be versatile, and many other languages may be used with it. Visual Studio and Eclipse are popular IDE options.  Available advanced server hosting options include website builder tools, mySQL databases, and unlimited Email accounts.    Linux app integration is complex and will likely involve the expertise of multiple teams.  KVM is a built-in virtual machine helpful in creating Window’s applications. | Visual Studio is a popular IDE for Windows as it was made by Microsoft. Eclipse is another good option that offers multiple languages.  Microsoft Azure is an available purchase option as a development tool for cross-platform mobile app creation and maintenance, storage and memory allocations, and reliable connectivity. Azure offers Linux virtual machine. | Android phones are creating from Linux while iPhone are created from Mac. Using similar IDE’s as Mac will provide most efficiency and compatitibility. |

## 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**: Considering the previous evaluation, I believe that using the most versatile Operating Platform for this project would be most suitable since the objective is to make the game available among multiple platforms. For this reason, I determined that the Windows operating platform would provide the most efficiency to host the game. There will likely be more experts available with the relevant experience needed to satisfy the requirements.
2. **Operating Systems Architectures**: Windows operates on a multiprocessor computer, designed for symmetrical multiprocessing. This increases CPU execution efficiency and reliability.
3. **Storage Management**: Windows is typically associated with more and cheaper storage options and hardware can be easily upgraded. Microsoft Azure is also available as a cost-efficient and secure cloud computing platform.
4. **Memory Management**: Windows OS uses a memory management system that directly contributes to the efficiency of the programs execution. Windows uses file mapping to associate the content to the virtual address as well as sequential I/O, both of which are known for higher process efficiency. With this game having a large library of images, both of these techniques will likely be useful.
5. **Distributed Systems and Networks**: Common distributed systems today involve operation on the web. Cloud-based servers have become a go-to for assigning workloads across processors. As mentioned previously, Window’s Azure is a cloud-based server that is a cost-efficient option capable of handling large amounts of data which can be accessed by other computers within the same network. This allows for a sharing of information of a local point in which security and connectivity can remain constant.
6. **Security**: The newest version of Windows security, Windows Defender, is continually updated with advanced encryption and data protection to provide network and cloud protection to the server and users.