

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 | 01/ 21/ 2022 | Laura McAroy | Design Template The Gaming Room/ Draw It or Lose It |

**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 developing a web-based game *Draw It or Lose It,* which is currently only available as an Android app. The game needs to be available on multiple platforms. One or more teams, with multiple players on each team, will be given 30 seconds to guess the drawing that is selected from a large library of images. If the team does not guess the answer correctly in 30 seconds, each member of the other team will have an opportunity to guess the answer within a 15 second time limit.

## [Design Constraints](#_2et92p0)

* Ability to have one or more teams involved
* Each team will consist of multiple players
* Game and team names are unique and cannot be duplicated
  + - Users are allowed to check to see if a name already exists
* Only one instance of a game exists in memory at any time
  + - Create unique identifiers for each instance of game, team, or player

These design constraints are only for the development of the game. The Gaming Room has also stated that they need the game to be available on all platforms, a possible solution would be to create a code that has the ability to inherit other languages.

## [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)

From looking at the UML diagram, it can be determined that the Game, Team, and Player class all inherit from the Entity class, making Entity the parent class. This means that Game, Team, and Player will all inherit the name and id references from Entity. It can also be seen that the child classes and the GameService class have a relationship as well, as Team has an instance of Player, Game has an instance of Team, and GameService has an instance of Game.

**"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** | Runs solely on Apple software but is tightly integrated with Apple mobile devices.  Has the capability to run both Windows and Linux side by side with MacOS  Popular in web hosting, but more expensive than Linux and Windows. | Exponentially more secure than Windows OS, but more difficult to find applications to support web hosting.  Open source, making it extremely cost effective. | Windows is the most prominently used platform, meaning there is more software available for this system.  Malware and Spyware are much more prevalent on Windows systems. | The most popular devices used, therefore have a wider reach than the other systems.  High portability  Low security compared to other systems. |
| **Client Side** | More costly than both Windows and Linux systems, but integrates seamlessly with other Apple devices.  Moderate expertise required | Limited application selection but more budget friendly than both Windows and Mac.  Maximum level of expertise required. | Less costly than Apple products needed to run MacOS, but more costly than Linux.  Minimum expertise required, Windows is user-friendly. | Mobile devices, i.e., cell phones and tablets, have much smaller screens than laptops or PCs, so user interfaces need to be developed accordingly.  Much more portable than a laptop or PC. |
| **Development Tools** | Objective C and Swift are the primary languages used by Mac OS.  XCode is the IDE used to develop. | An open-source operating system enables most, if not all, languages to be supported.  Eclipse is the most prominent IDE used. | C++ is used to build many of Windows’ applications.  Visual Studios is the prominent IDE used. | Android mobile devices primarily use Java, while Mac devices use Swift. |

## 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**: Because Windows is so user friendly, requires minimal expertise to use, and has so many available applications and IDE’s that can be run on its OS, I believe it would be the best choice to begin development of Draw It or Lose It.
2. **Operating Systems Architectures**: Windows OS are modular, containing two different modes. Applications run in user mode and core components of the operating system run in kernel mode where they have unrestricted access to the system’s memory and hardware. The processor switches between the two modes depending on what type of code is running. The restriction of programs running in user mode adds an extra layer of security to protect the system from malicious attacks.
3. **Storage Management**: I would recommend using Cloud storage for the development of *Draw It or Lose It.* Microsoft Azure is one of the leading cloud storage services, and offers one of the lowest rates, next to AWS and Google.
4. **Memory Management**: Because *Draw It or Lose It* will contain a large library of images, it will use a lot of memory. Windows uses virtual memory to map the location of the physical memory stored on the RAM for faster run times. It also utilizes memory compression, once a certain percentage of the RAM is used, the OS will begin to compress memory files so they take up less space.
5. **Distributed Systems and Networks**: Kivy is an open-source Python GUI framework that I found that enables you to write one code in Python that is then capable of running on all systems, Android and IOS included. Although it requires large amounts of storage space, since it is open source, it would be cost efficient and only one code will need to be developed for all systems.
6. **Security**: Windows boasts “Zero Trust” security principles on their operating systems, including multifaceted authentication, least-privileged access, and operating under the assumption that a breach will occur (utilizing end to end encryption and protected privileged roles). Even still, Windows is not known for its security, so additional measures such as anonymizing user data, would be necessary to protect users.