

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 | 09/18/2021 | Briana Carlson | First revision of software design |

**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 wants to develop their game Draw it or Lose it into a web-based application. The game is currently only available on Android but will need to be available on multiple platforms. The game has 4 rounds at one minute each where drawings are rendered at a steady rate until completed in 30 seconds. If the team does not guess it within the time limit each remaining team will have 15 seconds to offer one guess.

## [Design Constraints](#_2et92p0)

* The game needs to run on multiple platforms
* Only one instance of the game can exist in memory at any given time by use of unique identifiers for instances, teams, or player
* The game will need the ability to have one or more teams and each team will have multiple players assigned to it. The game and team names must be unique and allow users to check if the name is available when choosing a name.

## [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 diagram demonstrates the Object-Oriented Programming principles used throughout the program. This setup allows one game instance to go on at a time with multiple teams and multiple players. The entity class is the parent class that will provide base variables for ID and name for the child classes Game, Team, and Player to inherit from. The GameService class uses encapsulation with accessors and mutators for the game to hide the inner workings of the class but still show necessary information publicly. The GameService class has an association relationship with the Game class and the Game class, Team class, and Player classes are also associated with each other. The game is executed through the ProgramDriver class which uses the SingletonTester 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** | Mac has more flexible terminal commands to configure the server, to access, and make changes. | Linux also has flexible terminal commands and more open source and cost-effective advantages over the other options. | Windows has more software available. Windows is a more widely used operating system so more users though cost would be higher. | Mobile devices would be lacking in computing power compared to any regular computer though it would likely be more cost effective. |
| **Client Side** | While the cost would be like Windows, depending on existing expertise, it would require more time to gain experience as it is less user friendly | Cost would be very low as it is open-source software. It would likely require the most time, Linux is not as popular or widely used so it could be more difficult finding software developers for Linux. | Windows requires low expertise and time, more users are experienced with Windows than Mac or Linux, cost is similar to Mac | Good flexibility and low cost, experience would be less of an issue as they are the most user friendly. More time could be required as there are different operating systems and a variety of different device types. |
| **Development Tools** | Swift is the most common programming language for applications on Mac it’s easy to use and open source. There are tools in the safari develop menu and the Xcode IDE. | Linux has numerous free IDEs like Eclipse and Atom with open source developer tools like ren’py and Twine. It uses languages like C/C++ and Java | Eclipse and Visual Studio are very popular IDE’s for Windows with languages like HTML, Javascript, and C# | iOS like Mac primarily uses Swift. React Native is an open source mobile app development tool using Javascript |

## 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**: I believe that Windows would be the best Operating Platform to use for the expansion of their game. It is the most widely used platform with less time required to learn. There are many tools and languages to choose from when it comes to the development.
2. **Operating Systems Architectures**: The architecture of Windows enables applications to use kernel processes without directly affecting them. They could utilize the power of Windows for a GUI/window setup, memory and file management, or other vital processes to fit the needs.
3. **Storage Management**: Windows has built in features like Disk management and there is Windows Storage Management Provider designed for developers.
4. **Memory Management**: Windows uses built in memory management to monitor usage and uses a technique for memory compression once the available RAM has reached a certain point to free up more space.
5. **Distributed Systems and Networks**: Distributed Systems and networks allow communication with each other and differentiating processors across various platforms. Users would be enabled to communicate with different servers for web or data. If they’re on the same network, they’re able to run more efficiently since processes are split between the user and the server.

Distributed systems and networks work toward allowing communication across platforms. Users would have the ability to communicate across different servers

1. **Security**: Windows defender is a security function that can be used to encrypt data, it uses firewalls and antivirus to protect user information.