

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 | 01/16/2022 | Tim Brady | Completing an outline of the client’s requirements |
| 2.0 | 02/05/2022 | Tim Brady | Operating platform evaluation |
| 3.0 | 2/19/2022 | Tim Brady | Recommendation for architecture |

**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)

**Client:** The Gaming Room

**Request:** The Gaming Room is looking to expand their application “Draw it or Lose it” to multiple platforms using various software patterns in a distributed environment. An evaluation of operating systems is needed to make an informed decision.

## [Design Constraints](#_2et92p0)

* Must operate on multiple platforms
* User authentication security measures must be considered

## [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 ProgramDriver class contains the main and uses the SingletonTester to ensure only one instance of the game exists at a time. The Game, Team and Player classes all inherit methods from the Entity super class. GamesService, Game, Team and Player have “has a” relationships because a team has a player, a game has a team and a gameservice has a 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** | Upgradeable  Various options for web hosting requirements  Mac OS server available | Cost efficient  Most secure  Least used of the three OS so finding experienced people may be hard | More costly  Offers Windows server  Most used OS so may be easier finding experience. | Cost effective  Wide reach with high portability  Multiple OS means multiple more considerations |
| **Client Side** | User needs an Apple product which are costly. | Minimum cost.  High customizability.  Less users feel comfortable | Widely supported.  User comfortability. | Devices and OS range widely  Allow users to access almost anywhere. |
| **Development Tools** | Supports a wide variety of programming languages (Python, Java, HTML). Mac OS uses Objective-C  Swift and notepad++ are common | Supports a lot of the same popular languages.  Visual studio and eclipse | Supports a wide variety of languages. Windows uses C++ | Can use a variety of languages.  IOS uses Swift, Android uses Java. |

## 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**: My recommendation for The Gaming Room is to use Windows as there is a great deal of support and user’s generally feel more comfortable with the Windows OS. Windows offers a secure and convenient option for the cost.
2. **Operating Systems Architectures**: Windows offers ease of use with a simple GUI. There are many different software packages available and provides a wide variety of options in terms of programming languages. A command prompt allows the user a quick way to configure settings. There are two main components (kernel and user models). Kernel offers a way to utilize the power of Windows to access the necessary processes for the application without affecting other processes within the operating platform.
3. **Storage Management**: Windows offers functions like disk management and storage sense that allow a user to monitor and maintain file storage. Cloud server space is available for additional storage requirements.
4. **Memory Management**: Like the storage management, Windows allows users to monitor memory. Physical as well as virtual space exists for memory. Windows offers memory compression and a file system that will temporarily store pages when the app exceeds the RAM. A database would be needed for the game’s many image files for easy access.
5. **Distributed Systems and Networks**: Using a client-server system, we can have each application depend on a single server application. This would require a strong server network but would allow each client application to be developed to fit a client’s system.
6. **Security**: Windows contains some security protection software like Microsoft Defender that does things like scanning for malware. Regular updates to the system helps to ensure new attacks are kept up with. Additional software is also available. Extra security can be taken in the form of user authentication and need-to-know access.