

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01/23/2023 | Anthony Cunningham | In depth description of requirements and challenges of creating the game on all platforms as well as details on the preferred operating system to create the program in |

**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” has an android game called “Draw It or Lose it” that will render images from a library of stock drawings as clues and people must guess what is being drawn. This game must consist of 4 rounds each one minute long where the drawing is complete by the 30 second mark. If the drawing is guessed incorrectly the remaining teams must be given a 15 second opportunity to try one guess of the drawing themselves. They want this game to be developed as a web-based game. Each team, game, and player must have unique identifiers for each instance and done in cross platform code as much as possible.

**Requirements**

1. Ability to have more than one team involved
2. Each team can have multiple members assigned to them
3. Game and Team names must be unique. And should allow a check to see if a name is currently in use.
4. Only one instance of the game should exist in memory at a time

## [Design Constraints](#_2et92p0)

The gaming room wants this application to run across all devices which may require different code rewrites or cross platform languages like java to be involved for each device. Singleton Pattern was used in previous code, iterator class must be implemented moving forward to create only one instance of the games, team names, and players.

## [Domain Model](#_8h2ehzxfam4o)

The UML diagram below shows an outline of how the code will work for your game application. It displays the relationships between the multiple classes. We can start with the entity class. This can be viewed as a super class where Game, Team, and Player inherit from it. The entity and game service class also display signs of encapsulation to protect data from public methods. GameService > Game > Team > Player, have a “Has A” relationship also considered aggregation where one instance of the object in one class has a reference to an instance in another 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 a pliable or flexible system of commands to configure and make changes | Open source so many tools are free and has easily adjustable security setting. Migration could take a while. | Consistent and flexible resolutions. Average security and costly hosting. | The specifications vary on the operating system of the device. Inexpensive and a web app instead of a downloaded app can update itself but most likely cant work offline. |
| **Client Side** | Mac and Windows are more on the pricier side of things. A moderate amount of skill and time required for development | As Linux is less common a very high amount of skill is required to implement due to lack of templates and other resources. But as an open source minimal cost required. | Just like Mac as a fellow industry leading OS. The expertise and time are at a minimal due to all the available resources but cost at a high. | Maximum amount of time and skill are required to support the multiple mobile devices OS platforms |
| **Development Tools** | Most commonly used languages include but not limited to Swift, JavaScript, CSS, Ruby, Python, and HTML.  Written in IDE’s like Eclipse or visual studio. With tools such as, FlexiHub, Homebrew, and Xcode. | Linux can also use IDE’s notepad++, Visual Studio, and Eclipse. Languages include JavaScript, HTML, Ruby, Java, Python, CSS. The tools are frequent and free as Linux is an open source Operating System made by and for developers. They can include Autoconf, Bison, and Bluefish | Popular IDE’s from windows I have found besides the two basic one of visual studio and Eclipse can also include PyCharm, notepad, and even Microsoft word. And windows can run lots of languages as well like C++, C#, CSS, JavaScript, Java, HTML. Some tools for windows include Atom, Delphi, and Cloud 9. | Tools include ionic, Xamarin, Android Studio, and corona labs. While popular languages to code in for mobile app development can include Javascript, React Native, and Flutter, Swift, and objective C. They include Native languages and hybrid languages. Best IDE’s can include Android Studio, Xcode, or Visual Studio. |

## 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 would recommend windows. It has an incredible amount of software options ranging from languages and IDE’s as well as tools and libraries. Making actual skill as minimalistic as possible.
2. **Operating Systems Architectures**: A hybrid Kernel that comes in 32- or 64-bit architectures. That include a protected and supervisor mode, Program Execution, Memory management Virtual Memory Multitasking, Error detection and handling, Handling I/O operations. Resource allocation, as well as allowing disk access and file systems Device drivers networking Security.
3. **Storage Management**: Windows has a very simple file management system in place to let you go through all your drives and even has a new thing called storage sense to help free up space
4. **Memory Management**: The memory manager implements virtual memory, provides a core set of services such as memory mapped files, copy-on-write memory, large memory support, and underlying support for the cache manager.
5. **Distributed Systems and Networks**: Different connections like sockets or pipes may be used. Although errors like a lack of global clock or worn port info can cause connection issues. From a developer standpoint maybe a language like c++ or java that can be used to create software for all operating systems could be best used to make these errors as little as possible so that the code will only have slight adjustment for the individual OS’s.
6. **Security**: Windows comes with some preinstalled Security features although you probably definitely want more 3rd party applications for a professional environment like a VPN and other encryption software for the program data. The default windows security continuously scans for malware, viruses, and security threats. Updates are downloaded automatically on Windows 10 and 11. Windows also has “S mode”. This mode is streamlined for tighter security and automatically prevents viruses and other threats from running on your computer.