

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 | 12/15/22 | Joshua Morrow | Original Version |

**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 looking to create a web-based game that runs on a variety of different platforms. The name of the game is called “Draw It Or Lose It” and as of right now only works for Android. The game works by having multiple teams that have groups of people that go four rounds that lasts a minute. A picture gets pulled and shown to the teams and one team will guess what it is until the time runs out. If they do not guess correctly then each opposing team

## [Design Constraints](#_2et92p0)

-The game requires multiple people

-The game requires more than one team

- The game needs unique names for the teams

- Only one game can run at a time

-The game must be able to run on multiple platforms

The constraints of this game give us a guideline on what will be necessary to accomplish to complete the project as the company envisions. The game will need to run on more than just Android. This means that the code will either need to be changed to accommodate other platforms such as Apple or Windows, or will need to inherit the other languages so that the already written code can be ported over.

## [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 Entity brings in a relationship between the Player, Team, and Game class. Entity gives each of these some form of information. With the UML we show this by inheritance. Each of those three classes will have common references that are in Entity like “name” and “id”. We also see that GameService has a reference of Games, Games has a reference of Team, and Team has a reference of Player. Also, we see that Team and Player have a type. Game has a Team, and GameService has Games. In UML this is called aggregation.

**"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 the ability to be upgradable, and is popular in web hosting. However, it is less preferred than other types of web hosting services. | Linux is more cost friendly and is known to be more secured. However it is more difficult to find applications that support the hosting needs. | Windows is very compatible and has many different types of software. It has less loading time and high comfortability. However it is more susceptible to viruses. | Mobile devices server side have high portability and a wider reach. They also have better compatibility compared to most, but they are highly selective to the mobile devices they run on, and usually have poor security. |
| **Client Side** | Mac requires a moderate level of expertise to be used, and the cost is similar to platforms like Windows. A moderate amount of time is required as well. | Linux requires a high amount of time and expertise due to its complex nature. The cost is minimum however as it is open source. | Windows requires the least amount of time and expertise. The cost for this is similar to platforms like Mac. | Mobile devices require a low amount of time and expertise, and provides flexibility for seeing updates at any place. |
| **Development Tools** | While Mac can run just about any language out there. The main ones that would be used are swift and notepad++. | Linux is easier to run with programs such as visual studio, eclipse, and notepad++, but can run almost all other languages as well. | Windows is easier to run the different languages than the other platforms, but is similar in the ones that are best run on it. | Many of the apps that are run on mobile devices come from android and swift. The languages that they use are usually HTML, Javascript, and CSS, but are able to be run using other languages. |

## 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 think to start off the Gaming Room should use Windows as it would be the easiest to create the game for and most well known for all different forms of troubleshooting.
2. **Operating Systems Architectures**: Windows gives you a variety of services that is used by all other applications that let you see a GUI which accessing system resources. It also allows you to use the services through a user account or a server.
3. **Storage Management**: There is an application of Windows called storage sense that will not only allow you to use it as storage, but will track the stuff you put in there to show you how much space is being taken up. It also will also you to organize it so that the files and locations for apps are easy to find.
4. **Memory Management**: Memory allocation for Windows allows for easy storage of pictures outside of the default picture folder. This allows you to have the project together in one secure section of your computer.
5. **Distributed Systems and Networks**: There are programs out there such as Develop 4 which will allow you to create things like games with cross functionality enabled. This way whenever you finish your game you will be able to export is into a file that will go onto the web, or IOS, or Android, and will convert into a usable format on that platform. One thing to keep in mind with creating something that is multi-platform is that you will need a server that can handle large amounts of volume for players.
6. **Security**: Windows itself comes with built-in security software but will likely not suffice and it is highly recommended to get security software from another source. There are many options for security software that is out there that can run with Windows and keep your information secure such as AntiMalware, ProTools, etc.