

The Gaming Room

# **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 | 1/21/23 | Dylan Kusick | Executive summary, requirements, design constraints, domain model and recommendations have been updated. |

**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 a web-based game for many platforms. The game is called Draw It or Lose It which is available for Android only right now. The team will need help designing the game environment for multiple platforms with the design constraints listed below. This enhanced application will need to render stock drawings from a library rather than having players drawing them for clues.

## Requirements

* More than one team
* Each team will support more than one player
* No duplicate names

## [Design Constraints](#_2et92p0)

* One or more teams per game
* Multiple players on each team
* Both game and team names should be unique so there are no duplicates
* Only one instance of the game can exist at a time.
* Unique identifiers should be made for each instance of game, team, and player

## [Domain Model](#_8h2ehzxfam4o)

The Game, Team, and Player classes are brought together by the Entity class. This means that the three classes are child classes and Entity is the parent class that the other classes inherit from (represented by the open arrow). The UML diagram also shows there is 0 or more Players in a Team, 0 or more Teams in a Game, and 0 or more Games in the Game Service.

**"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** | Stable and reliable. These servers are great if you need to make sure you will not have a lot of downtime on your server. | Performance and open source are the biggest reasons to have Linux server side. It is nice to be able to have scalability to unlock the full potential of your server. | Easy to use with other windows products. There is loads of software you can use in tandem server side that will help you achieve high performance. | The biggest advantage is that they can go anywhere and still be functional. You can connect to the internet or access your location anywhere because of things like GPS and Wi-Fi. |
| **Client Side** | Languages like Swift, Objective-C, and Xcode may be needed client side as these are the languages used by Apple. | The open-source aspect makes it appealing to many that are knowledgeable of the operating system, though not as many people use Linux as Windows and Mac. | A lot of people have Windows so while it may be more costly than Linux it also is the most familiar to many people. | There are tons of different operating systems and versions of those operating systems that may end up costing a lot of money. |
| **Development Tools** | Swift, Objective-C, Cocoa, Carbon and Xcode are used by Mac developers. | Eclipse, C++, and Python are used by Linux developers. | Visual studio, C#, and the .NET Framework are used by Windows developers. | Xcode, Java, and Swift are used by mobile developers. Notice how there is a similarity to Mac due to the iOS for the iPhone and iPad. |

## 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**: In my opinion, Windows is the best operating system that can be used to help expand Draw It or Lose it. It is a widely used operating system with tons of flexibility.
2. **Operating Systems Architectures**: Windows has loads of great things to offer here as the gaming and software options are plentiful.
3. **Storage Management**: Windows 11 has a feature that allows you group 2 or more drives together just in case one of them fails so that you have a backup. You can add as many drives to this storage pool as you feel necessary.
4. **Memory Management**: The systems memory allocation and processes are tracked on Windows to keep your RAM running as efficient as possible
5. **Distributed Systems and Networks**: Unity can be used as it is a cross-platform game engine. Many independent game developers use it because it is simple. Unity is also great for 2D development and can be useful for mobile applications. The engine is used by many developers which means there are tons of libraries that can be used for the game.
6. **Security**: Windows has built-in security which means that you can scan for viruses and threats for free.