

Draw It or Lose It

# **CS 230 Project One**

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 | 03/16/22 | Brandon Kelfstrom | Added project information, system decisions, and UML diagram explanation |
| 2.0 | 04/01/22 | Brandon Kelfstrom | Added more information to evaluation |
| 3.0 | 4/16/22 | Brandon Kelfstrom | Added more information to Recommendations |

**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 needs a web-based version of their existing game “Draw It or Lose It.” The web version needs to support multiple different platforms. To accomplish this, the game will be hosted on a server. The website will detect the operating system of the user, and adjust display ratio accordingly.

## [Design Constraints](#_2et92p0)

* The game must be multiplayer, internet must be able to handle
* Game will have a team assignment system
* Games must be well synced for time restriction across client interfaces
* Each game will have one instance regardless of number of players to minimize memory usage on the server side
* Team names must be unique within each game, software must check and ensure this
* Game will be a static framerate, which can be decided later, but will most likely be 30fps for server-side optimization as well as client-side satisfaction

## [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 class inherits the Game, Team, and Player classes. There are 0 or more instances of the GameService, Game, Team, and Player linked respectively. The Entity class is a base class that holds objects for the 3 classes to use without declaring them in each class. This is an OOP principle called inheritance. The GameService class is not inherited by the Entity class as it is code used to create the game and not an in-game entity. The ProgramDriver is directly associated to the SingletonTester.

**"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** | MacOS is not commonly used as a server as the OS is exclusive to Mac computers. The hardware is usually not upgradeable or serviceable, and there are no rack mount macs made for server room application. | Linux is a free OS that is open source. It is popular for servers because it is low maintenance and low cost on ram. Many free and open source plugins are available for Linux to improvement functionality. | Windows is common server OS, but is prone to memory leaks. It has a huge customer support team, so fewer employees could maintain the server. The software is also designed with security in mind, but does cost between $500 and $1000 per system. | There is no reason nor ability to use mobile devices as the server. |
| **Client Side** | Mac applications must be developed in Xcode on a mac device. There are many mac developers as it is a popular system, so the cost of hiring extra developers for the platform should be worth it. | Linux is not very common as a PC OS. The system is widely understood by developers and it can run most programs available on windows. Development should be cheap enough even with a low user base. | Windows is the most common PC OS, and programs can be read in a variety of languages. Windows development is common and should be easy do | A huge portion of the human population has mobile phones. iOS and Android use different source codes, but because of the huge user base on each system, development is well worth the cost. |
| **Development Tools** | Xcode is the most common IDE for mac applications, but with special plugins mac code can be written in visual studios. | Visual Studios is a common IDE used for windows, but because of Linux’s versatility it can open programs developed in the main languages supported by visual studios. | Visual Studios is a popular IDE for development of code for windows. There are a huge number of development tools for windows since it is the most common PC OS. | Android studio for android devices, XCode for iOS, or Flutter in Android studio for single development for both devices, though it may decrease quality on both |

## 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**: Mobile platforms will allow the most growth because of the enormous user base. To cover most phones, an iOS version will need to be created. Creating a windows version will also capture a large portion of the market, allowing people on their computers to play the game.
2. **Operating Systems Architectures**: Android uses Java and iOS uses Xcode. These two platforms are somewhat different for development. The java version for android should be relatively simple to port over for use on windows.
3. **Storage Management**: Little information should be saved on the user’s device, but the default file system on the mobile devices should suffice. Any other information should be accessed from the server and transferred during runtime over the internet.
4. **Memory Management**: Memory will be allocated per game, storing the team names, team members, and other game/user information.
5. **Distributed Systems and Networks**: The game on the client’s device will communicate with a host server via the internet. The game’s functionality will be dependent on server up time, internet availability, and if the clients’ applications are still running.
6. **Security**: The client and server sides will encrypt data for transmission between each other. The key will be stored on the server side to prevent data breeches from users.