

Draw it or Lose it video game

# **CS 230 Project Software Design Template**

Version 1.1

## 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 | 03/20/2022 | Raven Oyunjargal | Added new information on all sections |
| 1.1 | 04/03/2022 | Raven Oyunjargal | Updated Evaluation table |
| 1.2 | 04/17/2022 | Raven Oyunjargal | Updated Recommendations for Draw it or Lose it |

**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)

A web-based application can be reached by more customers than software or app based applications. However what game features can be implemented can be very limited due to browser complexity and cookies. This can be solved by using JavaScript and React framework. React is flexible and can easily customize variables and features while game is running to accommodate game features and functions.

## [Design Constraints](#_2et92p0)

Due to web-based environment, implementation of features of the game will be slightly more challenging and time-consuming. This game having it’s own only one instance can be easily broken by running the game page in different browser tabs.

## [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)

ProgramDriver class used SingletonTester to identify if only one unique instance of game is active at a time. GameService class has zero to many relationship with Game as Game class a whole, has lists of gameService objects in it containing all GameService variables and functions available to call. Player as an object is part of Team class objects since few players compose a team.

Game, Team and Player classes are giving inheritance to Entity class who uses members and methods of these classes to store one game instance and game progress. It is required to access lots of Game, Team, and Player variables and methods.

**"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** | Thanks to it’s Unix underpinnings, Mac OS is excellent for developing web-based applications with built in terminal providing easy access to several development tools. However Mac is only on Apple units which are often more expensive. | Linux is user-friendly, well-designed and convenient for web-based development. | Same as Linux, Windows is also friendly, flexible with many programs and tools available than it’s competitor Oss and comes with variety of different hardwares and softwares. | Mobile devices are more limited in hardware and is not optimal for developing and running web-based application. |
| **Client Side** | Software development time vary but mostly stays reasonable within user’s timeframe. Cost however could be more expensive and not ideal for customers within budget due to Apple products prices. | Linux is optimal and user-friendly OS for development. It can take more time to get more developers and roles to get used to it as most people are only familiar with Windows or Mac OS. | Windows is considered optimal for most developers and testers as most of knowledge, tools and support comes within Windows OS. Budget and timeframe can stay within user’s expectations and requirements. | Mobile devices are not optimal for web-based applications as they require lots of resources and these devices are not manufactured to perform these tasks. |
| **Development Tools** | Languages consist of JavaScript, SQL, Java and C++. Visual Studio and Visual studio code and Eclipse. | Languages on Linux consist of but not limited to:  C++  Python  Shell  Perl  Google Go.  IDES in this OS are:Eclipse and CodeLite. | Windows is capable of running most IDEs and languages compared to other OS.  C#, Python, Java, JS with frameworks React, Angular and Django for web development. | Mobile devices often use Java and C++ to develop apps. |

## 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**: Recommended operating system is Windows OS. It is used and familiar with more developers and hosts most tools and support to develop and keep web-based application running.
2. **Operating Systems Architectures**: Windows OS has different editions, but important factor is 32 bit and 64-bit systems. 64-bit systems perform better than 32-bit counterpart and more recommended.
3. **Storage Management**: SSD loads faster than traditional HDDs. It is more optimal choice for web-based application as it allows it to load faster and to more users at a time. SSD’s storage is lower than HDD, so it is important to keep file sizes low and in batch to save space.
4. **Memory Management**: Windows OS has huge memory capacity and can also use page files as dedicated memory to save memory for application. It has 4-8GB of virtual address and all threads of process can access virtual address. Threads can’t access memory that belongs to another process which protects process from being influenced or changed by another process as it is important for all instances of Draw it or Lose its integrity.
5. **Distributed Systems and Networks**: By using cloud storages and server’s user won’t have to buy traditional dedicated server computers and keep them running. Azure, IBM and Google clouds are efficient choices. Traditional onsite servers are more costly and require maintenance.

Azure has several regions with each region has multiple data centers spread throughout. In case of server outage and loss of files Azure can help recover files or issue refund for damage caused.

1. **Security**: On this section Linux is more secure and less prone to attacks than its competitors. Due to how file system is structured, and user account restrictions Linux is better protected against malware attacks.

On other Operating systems:

Windows OS – User can use antiviruses and regular OS updates to be protected against threats. Also restrict all incoming and outgoing network traffic through firewall. Similar to Linux, you can create accounts with restrict them with privileges.

Mac OS – Security features include firmware password, Find My Mac to track, locate and wipe device, Apple ID Two-Factor authentication for system access and System Integrity Protection for system files integrity.

Mobile device-

* User can install anti-virus app to protect against malwares
* Set secure password
* Keep phone OS updated
* Only connect to secure Wi-Fi.