

Draw It or Lose It Environment

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 11/13/2022 | Wesston Reed Mccollum | Revised previous version to add abstraction and inheritance |
| 1.1 | 11/27/2022 | Wesston Reed Mccollum | Revised Evaluation and Recommendations with more researched information |
| 1.2 | 12/10/2022 | Wesston Reed Mccollum | Revised Recommendations with more researched information |

## [Executive Summary](#_sbfa50wo7nsh)

This project sought to develop an environment for The Gaming Room that would run their web-based game Draw It or Lose It. This concept was also accompanied by requirements of:

* One or more teams per game
* Multiple players per team
* Unique game and team name verification
* One instance running at any given time

To accomplish these tasks, several computer science concepts were used such as abstraction, inheritance, and singleton pattern design.

## [Design Constraints](#_2et92p0)

The game must allow for any instance to only reference itself. Without this cross referencing can occur that could lead to systemic failure. This can be prevented by using singleton pattern design. This design can also aid in restricting the search query range of names and identification numbers for games.

## [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 main information holding classes shown are Entity and GameService. GameService is used to initialize a game and its information. Entity is a parent class that is used as a template for similar information of Game, Team, and Player classes. Each of the three aforementioned classes inherit from Entity and use it as abstract to do their individual functions.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac usually offers a more secure hosting environment, but it is hard to upgrade; strong option: Mac OS X, $999 licenses as well as the investment into hardware | Very customizable but can be less secure without much support if done independently; high variety such as Ubuntu, Red Hat, etc.; costs would include cloud server hosting or can buy licenses through certain Linux systems | Windows is widespread with capabilities to upgrade but can be prone to attacks; Azure PlayFab can cost depending on the number of users you receive or be standardized | The most used type of device in the world but needs very specific implementation; no specific hosting method, derived from other platforms |
| **Client Side** | Would potentially need a Mac specialist for admin operations, but much support has been discontinued as of April 2022, would need special tailored integrations | Would be highly customizable, allowing for extreme accessibility, diverse code integration | Development would be most prevalent for Windows with widely diversifiable application; MacOS and iOS would require | Requires very specific implementation, potential for development if mobile specific site, can be the most used device. |
| **Development Tools** | Swift is the most prevalent language used in Apple products; has Mac OS X, while discontinued still offers a UNIX based server option; dev would require Mac machine | Can be integrated to most any language; can be used virtually | Python and Java are the main languages preferred for Windows; can be used without dev tool costs | Swift, Python, and Java can be seen most prevalent; some free dev tools, would be largest market for most used device |

## Recommendations

1. **Operating Platform**: It is recommended that Linux be used as the Operating Platform for the server desired.
2. **Operating Systems Architectures**: The flexibility to have either a graphic interface or command line shell allows for variable use across all needs. Also, Linux OS is completely open source, meaning the architecture can be changed to meet most any requirements.
3. **Storage Management**: Devices can be added and removed from the base file system of Linux very easily, and sorting information revolves around partitioning from parent devices.
4. **Memory Management**: Linux relies on virtual memory, which takes memory addresses and translates them into physical memory addresses that can be accessed. This is optimal for high-volume interactions, such as the volume that would be involved with Draw It or Lose It.
5. **Distributed Systems and Networks**: The Linux kernel is open source, meaning it can be altered to match necessity. There are many Linux distributions such as Red Hat that are comprised of package management systems, integrated software, and the Linux kernel that offer scalability and redundancy for maximum server efficiency.
6. **Security**: At its most basic, Linux OS offers great protection through clearly defined privileges that restrict access to the system’s root from outside networks. Also, Linux defines every connection and device as a “file” meaning easily definable access restriction.