

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 | 10/02/2022 | Eyoel Tesfu | Added Evaluations and OS Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The game called “Draw It or Lose It” is already available on the Android platform. Requirements include the creation of a web-based game that mimics “Draw It or Lose It” and its launch on multiple platforms.

Components of the “Draw It or Lose It” game include:

* It is a team-based game
* An automated application will randomly pull and slowly render clues of an image selected from a huge library of stock drawings, the teams’ task is to guess the image correctly
* One game consists of four rounds, which are one minute each
* In every round, drawings of the randomly pulled image renders at a steady rate. Render of the image will be fully completed at the 30-second mark.
* If the team does not guess the puzzle before time expires, each remaining team will have an opportunity to offer a guess to solve the puzzle with a 15-second time limit.

## [Design Constraints](#_2et92p0)

* The code should include a web-based language such as Python or JavaScript
* Game names must be unique
* Only one instance of every unique game can exist in memory at any given time. The creations of games must therefore be based on a singleton pattern
* Users will have to check if a game is in use by using its unique name
* A game requires one or more teams
* A team will consist of multiple players
* Team names must be unique
* Player names must also be unique

## [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)

* Entity is a super class. Game, Team and Player classes inherit from Entity class. This means that the games, the teams, and the players are all entities.
* An Entity will have an id and a name as attributes, which means that each game, team, and player will have its own id and name.
* A game will have a team (or teams), which have players.
* Every game also will have a GameService. There will only be one instance of GameService since it is designed in a singleton pattern. Therefore, that one GameService instance can serve multitudes of games.
* Encapsulation is used by the GameService, Game, Team and Entity classes as they limit access to their private attributes from other classes. This will also be useful for easy code testing and reusability.
* ProgramDriver includes main() and is the main framework package that runs the program. It also cans use SingletonTester class 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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac computers are relatively expensive. Has an easy method of server-based accessibility and configuration that can be done through a workstation. Mac based servers serve only Mac users, so there might be incompatibility issues. Not recommended since application needs to be on multiple platforms. | It is cheaper. It is the most popular choice for web servers. It is more difficult to configure and customize since it needs extensive knowledge. It is the more popular choice for server side as it requires minimum resources. It has better security with great community support. | It is more popular and user friendly. It is the second most popular for web servers. It however has more security threats due to widespread use / ubiquity. | It is cost-effective. Specifications will likely be way lower than required. Inadequate screen real estate might also limit the speed of workflow. It has its advantages but is not ideal. Most of these devices don’t have the necessary storage nor processing power. It is not recommended to use mobile phone devices as a web server host as they might also not support  Website files and databases. |
| **Client Side** | Expensive. Friendly GUI can increase interaction quality. Relatively quick and easy for the user to get used to the environment. Mac browsers can send to and receive data from the server via the REST API. | It is free to use since it is open source. More flexible handling of the GUI, customizability. User must be well versed with Linux. It is hard for new users to get accustomed to Linux. Can take more time to set things up. Linux clients can communicate with the server via a REST API. | The hardware varies in price, but the software is not too expensive. Most popular OS among users. It is very user friendly. Relatively easy to learn and get accustomed to. Windows can communicate with the web server via HPPT protocol using REST API. | Devices can be cheaper. These are even more ubiquitous than Windows PC devices. As such, a lot of users are expected to have an easy time navigating it. The server side must be able to receive HTTP requests from these devices. We can apply this by installing an HttpClient plugin and REST API on the web server application. |
| **Development Tools** | Most popular languages used to deploy software on Mac include Swift and Java. JavaScript, Python, HTML and CSS can also be used. IDEs include Visual Studio, Eclipse and Notepad++. | Can run all kinds of programming languages. Most used languages are C/C++, Java, Python, JavaScript, and Shell. IDEs used include Visual Studio, Notepad++, Git, Node.js … | Has a wide variety of software tools to choose from. E.g., Microsoft Visual Studio, Eclipse, Notepad++  Popular languages include JavaScript, Java, Python, C/C++ | Software can have restrictions on the development side on these platforms. Mostly used languages include BASIC, Kotlin, Swift.  IDEs include Android Studio, Apple developer tools, Eclipse, CodeSnack and some Text Editors. Since there are multiple types of operating systems, multiple development teams may be required to develop an application for each operating system. |

## Recommendations

1. **Operating Platform**: Windows, while being the most user-friendly, is the most versatile and requires less expertise compared to other operating platforms for development. Windows servers also have built in support for file servers, application servers, database servers, and web servers
2. **Operating Systems Architectures**: Windows provides services used by all windows-based applications and delivers data in a well-made GUI by accessing system resources in a conservative way. Web-services can be included as one of these applications. It’s high compatibility with other operating systems makes it a good candidate for development. The multi processors aspect of Windows also provide systematic scheduling. These systems can be accessed through an authentication panel for performance and safety.
3. **Storage Management**: Windows provides the necessary tools for flexible storage configuration and effective and easy cloud storage management and accessible upgrade. Primary concerns should revolve around virtualization, data compression for weaker client systems, data replication and migration for backups, disaster recovery tools, and automation of these processes.
4. **Memory Management**: Windows has its own virtual address space for each process, which allows up to 4 GB of memory to be viewed. Additionally, threads do not have access to the memory of another process. This helps in protecting one process from being damaged by another. It also allows for contiguous memory allocation while also freely disposing resources for both static and dynamic linking and loading. Memory management essential for minimizing fragmentation issues, maintaining data integrity in instances of process, and allocating and de-allocating memory before and after process execution.
5. **Distributed Systems and Networks**: The flow of data needs to be accessed by multiple platforms. The data that users can’t access because of incompatibility should have duplicates that provide compatible backup. Routing, congestion, and queuing problems can be mitigated by increasing the capacity (storage), having better methods of connectivity, closure of congestive networks, and having more advanced scheduling and synch able signals of distributed systems. Outages can also be covered by backup servers with backed up power.
6. **Security**: Windows comes with its own built in security software. The admin system also adds a layer to the security system. However, it is vulnerable since it has a huge user base. For this reason, it is a better choice to have extra measures of security from an outside party. Additionally, the encryption of data is a must as user information should be protected. Primary third-party candidate is Malwarebytes Premium.