

Draw It or Lose It

# **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) **and 4**

**Evaluation 5 and 6**

**Recommendations 7**

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/20/2023 | Courtney Warner | Submission #1 |

**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* is interested in developing a web-based game from their already existing Android game *Draw It or Lose It*. They would like the game to be available on multiple platforms. Operating on multiple platforms, they will be able to generate more users and increase profits.

## [Design Constraints](#_2et92p0)

* Needs to run on multiple platforms
* Needs to render images from stock images in a large library
* Game consists of four rounds that last one minute
* Drawings are rendered at a steady rate and are fully complete at 30 seconds
* If no correct guess is made before time expires, other teams can attempt to solve with one attempt and 15 seconds
* At least one team is needed to play
* Multiple players per team
* Unique player, game, and team names required

## [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 UML diagram represents the programming for *Draw It or Lose It*. It consists of seven classes. The classes and their general purposes are as follows:

* ProgramDriver class is the Driver class and contains the public main method
* SingletonTester class used to test the singleton following the singleton pattern, to ensure a single instance of the game is running
* Entity class is the base or superclass
* Game, Team, and Player classes inherit the attributes, and methods from the Entity class
* GameService, Game, Team, and Player classes all have a 0 to many association with each other
* GameService *has a* 0 to many association with the Game class, as GameService can have multiple games
* Game class *has a* 0 to many association with the Team class, as a single game can have multiple teams
* Team class *has a* 0 to many association with the Player class, as a single team can have multiple players

**"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** | **Advantages:**  \*Has a server-based deployment method  **\*** Flexible terminal commands  \* Security  **Disadvantages:**  \* Less gamers use Macs, so Mac software is not good for gaming  **Cost:**  \* More expensive | **Advantages:**  \*Has several server-based deployment methods  \* Flexible terminal commands  \* Security  \* Can run on old Hardware  **Disadvantages:**  \* Less commercial software available  **Cost:**  \* Can come with cost but can be free | **Advantages:**  \* Has a server-based deployment method  \* Steady Stable platform  **Disadvantages:**  \* Security issues  \* Frequent updates needed  **Cost:**  \* Less expensive than Mac | **Advantages:**  \* It is possible to use server-based deployment  \* Push notifications  \* Scalability  \* Reliability  **Disadvantages:**  \* Lack of support  **Cost:**  \* Relatively less expensive |
| **Client Side** | \* Requires more expertise than Windows but less than Linux  \* Moderate expertise means moderate costs for the company and a moderate amount of time  \* Compatible on most browser but best for Safari | \* High level of expertise needed  \* Complex installation, not user-friendly  \*High-level expertise will equal high-level cost and relatively more time to develop  \*Compatible with most browsers, however some features may not be available | \* Easy installation  \* Relatively easy installation should mean a moderate cost for development and a moderate time  \*Some browsers are not compatible, however, works with the most popular browsers | \* User-Friendly  \* Would require coding for Android and Apple users  \* Developing for both Apple and Android will take either a team that is able to develop both or two teams, either option is most likely to be costly  \*Most browsers are compatible on mobile devices however may not be optimized on mobile devices |
| **Development Tools** | **Programming Languages:**  Swift  HTML  Javascript  Python  Java  C++  **IDE:**  PyCharm = Moderate Cost  Visual Studio = High Cost  Eclipse = Free | **Programming Languages:**  HTML  Javascript  Python  Java  C++  **IDE:**  PyCharm = Moderate Cost  Visual Studio = High Cost  Eclipse = Free | **Programming Languages:**  HTML  Javascript  Python  Java  C++  **IDE:**  PyCharm = Moderate Cost  Visual Studio = High Cost  Eclipse = Free | **Programming Languages:**  (Depending on the mobile device)  HTML  Javascript  Python  Java  C++  **IDE:**  PyCharm = Moderate Cost  Visual Studio = High Cost  Eclipse = Free |

## 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**:

For *Draw It or Lose It* Windows operating system would be the best recommendation. Windows offers faster speeds and direct storage; both are important for gaming. Windows is also the most widely used operating system and therefore would open to a wide range of users.

1. **Operating Systems Architectures**:

Windows uses two modes/components the user mode and the kernel mode. The user mode manages processes for users. The user mode has fewer privileges/access to the system processes. This lessens the chances of system corruption. Windows can allow applications to switch processes from user to kernel when necessary and back when necessary. Kernel mode is the other component of Windows where the core system processes run. The kernel is very important to the operating system if the kernel mode driver crashes the operating system will also crash. Windows also uses two stacks, one for the thread/kernel mode and one for the user mode. As well as runtime and dynamic link libraries.

1. **Storage Management**:

Windows has several different ways to manage storage, they are Storage Sense, Disk Cleanup, and Disk Management. Storage Sense can automatically free up drive space. It does this by adding temporary files and not needed files to the recycling bin, which will then be deleted after some time. However, downloads and OneDrive are not affected. Disk Cleanup has a similar function as Storage Sense. It also helps clean up unnecessary files by deleting them. Disk Management helps with more advanced storage tasks like initializing a new driver.

1. **Memory Management**:

Windows has virtual address space for each 32-bit and 64-bit process. All threads have access to the visible address space. Accessible memory space goes equally to the kernel and user programs. Windows memory manager implements virtual memory and provides core set services, such as memory-mapped files, which can be used to share files or memory between processes.

1. **Distributed Systems and Networks**:

A client-server distributed system will be needed since Draw It or Lose It will be running on multiple platforms. Having a client-server model will help partition tasks. This will be a cost-effective option with low maintenance costs. Centralization, accessibility, management, and scalability are just some of the benefits of the client-server model.

1. **Security**:

Windows Security and Window Defender offer built-in layered protection. With Windows Security the user information can be protected with authentication factors such as a user password. In addition to these many anti-virus protection software options are available for purchase for additional security