

# 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 | 10/14/2021 | James Carver | Working game, team, and player classes  Add new class, Entity class |

**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)

Creative Technology Solutions (CTS) has recently taken on a new client, The Gaming Room. The Gaming Room wants to develop a web-based game that serves multiple platforms based on their current game, Draw It or Lose It, which is currently available in an Android app only. Draw It or Lose It is loosely similar to the 1980s television game *Win, Lose or Draw*, where teams compete to guess what is being drawn. Rather than a player drawing images on an easel to help team members guess the puzzle (a phrase, title, or thing), the application will render images from a large library of stock drawings as clues. A game consists of four rounds of play lasting one minute each. Drawings are rendered at a steady rate and are fully complete at the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

## [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 below has seven classes: programDriver, SingletonTester, Entity, GameService, Game, Team, and Player. Entity class is a parent class and has four child classes. The singlectonTester class inherits from programDriver class which is the main class. The OPP principle applied in this UML diagram is inheritance. It has enabled the singlectonTester class to perform activities and responsibilities inherited form the main class.

**"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** | **Characteristics**  It is popular in web hosting  **Advantages**  It is upgradeable, it has various options for different web hosting requirements  **Disadvantages**  It is less preferred for web hosting services | **Characteristics**  powerful, stable and reliable yet easy to use **Advantages**  Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  Linux systems are widely known for their stability and reliability  **Disadvantages**  It is more difficult to find application to support the web hosting required needs. | **Characteristics**  It is dominant to the other platforms.  Close platform  Windows hosting is a web hosting which locates user file on a web server that uses the windows operating system  **Advantages**  High resource requirements, less loading time, high comfort ability  It supports traditional hypertext mark-up language files.  **Disadvantages**  easy virus susceptibility, poor technical support | **Characteristics**  More popular, high portability.  **Advantages**  Has a wider reach, better compatibility, cost-effective  **Disadvantages**  It is highly selective to various smart mobile devices  Poor security  The hardest option is for you to code your entire backend yourself and find a provider that will host your mobile app. |
| **Client Side** | High expertise required to develop software for clients who pertain to Mac.  It is expensive, MAC is costlier than Windows and the user is forced to buy a MAC system Much time is required to access the software | Requires high expertise as few applications are available.  Less loading time.  Most of the software for it is a open-source | It requires a high expertise as it has high resource requirements.  Windows is a great platform for **developing** web sites and native **windows** apps using Visual Studio | Phones have very limited screen space, and tablets, while bigger, are still mobile devices with less screen space than even most laptops. Because of this, mobile platform UI controls have been designed specifically to be effective on smaller form factors.  It is cost-effective  It takes less time to load a page |
| **Development Tools** | **Objective-C** is the language most used in Mac OS Programming.  Also, PHP programming language  JavaScript | C, C++, CSS, Java, JavaScript, HTML, PHP, Perl, Python, Ruby, or Vala, Linux supports them all. | uses **C++** to build many of its core applications.  **C++** is widely used in the software industry, and remains one of the most popular languages.  Java and HTML/CSS are also used. | Swiftic is one of the **best mobile app development** frameworks available in the iOS platform.  **Java** is the official language for Android App Development.  Android Studio |

## 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**: Windows Operating platform is recommended as it is common in developing web-based software. It is highly secured with less loading time and is relatively cheap. It is portable and compatible therefore the right choice for the development of the Lose It or Draw It game.
2. **Operating Systems Architectures**: Windows architecture has 2 components: user mode and kernel mode. Kernel mode includes security. I/O management, memory management, processes and threads, and others to manage the backend of the program. The user mode give access to applications without the worries of system software or hardware.
3. **Storage Management**: Starting with Windows 8 and Windows Server 2012, Windows has Windows Storage Management API. This along with cloud-based storage will give the devs and the game plenty of space.
4. **Memory Management**: Windows applies memory compression technique. This will increase the responsiveness of the operating system. The page file system will start removing pages of memory out of RAM and store them temporarily on the hard disk when the amount of memory for the software exceeds the of RAM available.
5. **Distributed Systems and Networks**: A distributed system is a system with multiple components located on different machines that communicate and coordinate actions in order to appear as a single coherent system to the end-user. This increases stability, reliability, and performance. By spreading out the workloads, distributed systems can support more requests and compute jobs than a single system.
6. **Security**: Due to the high security capabilities for Windows operating platform, the user protection against intruders will be higher. Windows offers Azure Defender, Azure Sentinel and Microsoft 365 Defender to protect all aspects.