

# **Draw It of Lose It**

# **CS 230 Project Software Design Template**

Version 3.0

## 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)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01/21/2023 | Andrew J Cummins | This is the first version so no changes have been made. |
| 2.0 | 02/11/2023 | Andrew J Cummins | Edits have been made to the evaluation with specific regards to the requirements of the Gaming Room. |
| 3.0 | 2/24/2023 | Andrew J Cummins | Edits have been made to the recommendations. |

**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 has requested Creative Technology Solutions (CTS) to make a web-based game, which is currently only available on android that can run on multiple platforms. The game, called Win, Lose, or Draw, has four rounds all a minute a piece. Each round team members must guess a puzzle based on a library of images used as clues. At thirty seconds the puzzle will be full of all images. If at the end of the round the team does not get the puzzle the remaining teams will have fifteen seconds to solve the puzzle. CTS will provide the software design and development to meet Gaming Room’s request.

## Requirements

Win, Lose, or Draw needs to have the ability to have one or more teams involved and each team needs to have multiple players able to play.

Win, Lose, or Draw must have a database storing the team, game, or player names so that no duplicates arise.

Win, Lose, or Draw must work on multiple platforms such as Windows, Linux, and Apple.

## [Design Constraints](#_2et92p0)

The design constraints for Win, Lose, or Draw are mostly developmental. The existing code that has been designed for Android will need to be re-written in order to meet the demand of multiple platforms. The other design constraints are about the requirements of the game itself and

## [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 the parent class that creates a relationship with the Game, Team, and Player classes. The Game, Team, and Player classes “inherit” the attributes of the Entity class. The GameService class relates to the Game class and provides the access modifiers. The attributes are all private so that only one instance of the object can be called at a time by the ProgramDriver. The operations of the GameService class serve to add or call an already existing instance for the game, team, or player. The GameService, Game, Team, and Player classes demonstrate polymorphism, a form of abstraction, as each class relates to each other in that the operations to form the next class lie in the previous class. The SigletonTester class relates to the driver class so that only one instance will be called at a time in the ProgramDriver.

**"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** | Mac OS supports the deployment of gaming applications on a web-based server. Most popular for hosting web based applications however it is the least preferred.  Mac OS is particularly good at hosting multiple servers with different domains. | Very cost friendly and most secure platform because a lot of its software is open source. However, it has a small market and therefore fewer applications associated with it.  Although Linux accounts for a very small amount of Desktop interfaces it is the preferred OS for the web which is what makes it a compatible platform. | Most software and comfortable interface for end users. Poor security and technical support. The software for Windows can be very expensive when the licensing fees begin to add up. Windows works with Microsoft which is constantly being updated. Windows has a large memory management capacity | Mobile OS supports the deployment of gaming applications on a web-based server. It is also the fastest growing platform for gaming applications. The mobile platform has become the most Cost Effective and can reach more users due to its portability. It is difficult to hold a server if connection is lost in movement |
| **Client Side** | Mac requires a moderate amount of expertise and time required for software development. Has challenges using Windows data and can be very expensive for end users.  I found Mac to be the most versatile operating system in that it links well with the other three operating systems being analyzed | Linux requires the most expertise and time out of these operating systems for software development. Less understanding among end users and software compatibility. Most complicated interface for end users. Linux is open source software which allows anyone to work on your source code. This is good for security because many developers can work on patching code or preventing potential threats. Linux is said to be compatible with all naturally other than windows OS. | The least amount of expertise and time are required for software development. The amount of licensing agreements can be a downfall. User friendly interfaces and most used software. With the proper links Windows and Android are very compatible with one another. Windows requires a 3rd party link in order to join the Linux platform. | Mobile apps require a moderate amount of expertise and time required for software development. Fastest growing market. Constant updates and have to pick and choose what to use limited download space on. It is also found that almost ¾ of end users use an application for 72 hours and never use it again. Android is very versatile with all the other operating systems. |
| **Development Tools** | Mac supports all programming languages. Swift is a very popular language that is effective for implementing software. Hype is the best tool to create animations. Mac OS supports JavaScript which can be used to return HTML that will allow for the web based application to be deployed successfully. Developing Mac applications will require moderate amount of developers. Mac has the lowest licensing fees behind Linux. | Linux supports visual studio, eclipse, and notpad++. IntelliJ IDEA is accept as the industry standard. Linux supports JavaScript which can be used to return HTML that will allow for the web based application to be deployed successfully. Linux may require multiple developer teams due to the extensive time and expertise required to build. Linux has the cheapest licensing since most of the software is free. | I run windows operating system and I have used Eclipse IDE and PyCharms which use numerous programming languages such as python, java, and c++. Windows OS supports JavaScript which can be used to return HTML that will allow for the web based application to be deployed successfully.  Windows is an OS I presume would require multiple developer teams in order to get the software to make it compatible with the other platforms. Also the software comes with licensing fees. | Xamarin is the preferred developmental tool for native applications. Supports SQL for database management. Mobile platforms supports JavaScript which can be used to return HTML that will allow for the web based application to be deployed successfully. Less developers are required since mobile applications are versatile and the game is already built for android use. The use of emulators could speed the build for other platforms. The licensing fees to keep applications in the google and apple stores could begin to add up if you stop profiting. |

## 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**: I recommend using Linux operating system to make this Android based game into a web based game. Android operates of a Linux based system and therefore will be the most cost effective way to. Linux is also free to download.
2. **Operating Systems Architectures**: The architecture of Linux includes 4 main layers. The applications/utilities, the shell, the kernel and the hardware are the fundamental components of the Linux operating system.
3. **Storage Management**: I would recommend to outsourcing the storage to hard drives or a RAID. Disk Partitions are also very compatible with Linux OS.
4. **Memory Management**: Linux supports virtual memory which will in a way offload the memory onto a disk as an extension of RAM therefore leaves the OS more room to run the game for faster connectivity.
5. **Distributed Systems and Networks**: Linux is an open source OS which allows programmers the ability to design it for multiple operating systems. This allows any client to use the services and manage the hardware resources effortlessly.
6. **Security**: The least likely of the OS mentioned above to become susceptible to viruses. Linux is highly secured with a 3 step access process. Linux can use User name and Password, authorization and encryption to provide security.