

Draw It or Lose It

# **CS 230 Project Software Design**

Version 1.2

## Table of Contents

[**CS 230 Project Software Design**](#_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 | 07/30/21 | Sergio H. Passos | Initial design for the Draw it or Lose it game |
| 1.1 | 08/01/21 | Sergio H. Passos | Update and fixes |
| 1.2 | 08/12/21 | Sergio H. Passos | Update and fixes |

## [Executive Summary](#_sbfa50wo7nsh)

The company wants to develop a web-based game that can run on other platforms besides Android, such as iOS, Windows, Mac, and Linux. The company already has their game “Draw It or Lose It” on the android platform. The game consists of grouped up players partaking in four rounds that last a minute each. Images are pooled from a list and delivered to the players for them to attempt to guess the image. Then the counter group gets a chance if the other teams has failed top guess the image in time.

## [Design Constraints](#_2et92p0)

* The programming language
* Game and Team names must be unique
* Only a single instance
* Web-based delivery method

The language itself is probably its largest constraint, since the objective is to deliver the game from a web perspective. This programming language needs to fulfill the design by allowing the game to run and get delivered to clients on request. This leads to another constraint is how the network will handle users and their game from other users and their game.

## [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 ProgramDriver contains the main function for the program; this is the entry point. Additionally, it also uses the SingletonTester object. The core functionality resides within the GameService class. GameService has an association with the Game class. Additionally, its multiplicity is zero to many. Which means that the Game class can have as little as zero instance or a multitude of instances. The Game class also has an association with the Team class with the same multiplicity, zero to many. Lastly, the Team and Player class also have an association with a zero to many multiplicities. The Game, Team, and Player classes all inherit from the Entity class. The Entity class contains basic attributes such as an id, name, and functions related to those attributes.

****

## [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** | The hardware and software are limited for this operating system. However, Mac offers great server side and client-side communication between other Macs. | Arguably the most popular server-side operating system for its high customizability, compatibility, and stability. | This is also another popular operating system due to how common it is at home and work. This comes with similar perks as Linux. Compatible with allot of hardware and software. | This should be the last option for server-side hosting. P2P hosting should be the extent of mobile server-side hosting. |
| **Client Side** | Swift is MacOS native programming language. Apple supports this and offers a development client for developing apps. Additionally, this also translates well for all apps on Apple’s eco system such as iPad, iPhone, iWatch, Mac, MacBook, iMac, and MacBook Air. Apple does charge an annual $100 dollar fee to publish applications. | Linux would have to be the cheapest platform. Linux offers many development tools to develop applications, unlike Apple. This allows for a very highly customizable experience. | Windows like Linux has a multitude of development options. From Microsoft’s own Visual Studio to other paid and open-source options. Also, there is no cost in publishing an application. Additionally, publishing can be done on an ever-growing list of popular stores on the Windows platform. | Developing a web application for mobiles would be the easiest. Additionally, if you wanted more control and power than Java, Koltin, C#, and Swift would be the most popular options. C# can publish to both platforms simultaneously. While Java or Swift would have to be developed respectively for their own platforms, Android, and Apple. Many options exist for the mobile platform. |
| **Development Tools** | Swift would ultimately be the best language to develop applications with XCode. | Linux has plethora of languages to develop. From Python, C, C++, Rust, Ruby, Ruby on Rails, and much more. Also, the most popular development applications are GNOME, KDE Frameworks, and Elementary OS just to name a few. So, options are a plenty in Linux. | This is also like Windows. Windows offers their own language C# and tools such as Visual Studio to develop applications in. However, users have many options such as C++, Java, Python, Ruby, and much more. Additionally, other popular tools exist from paid to open source. | Java is the leading development language for developing Android applications, along with Eclipse. Kotlin is another up and coming language developed by Google to slowly replace Java. Additionally, is also can work with Java. |

## 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**: The Gaming Room should start with Windows since its extremely common and should have almost all development options needed to complete the project.
2. **Operating Systems Architectures**: Windows takes a layered approach; it has a user mode and kernel mode. It has a multitude of small modules that make up features in the OS. Modules such has Hardware Abstraction Layer, Kernel, Executive Services, Environment, and Integral subsystem. These all work in conjunction to bring a safe, reliable user experience.
3. **Storage Management**: Windows storage would work just fine. However, the team should setup a cloud-based storage system such as Google Drive, OneDrive, or Drobox. This solution would serve to reduce client storage use. Which is paramount in the development world since losing data would make it difficult to continue.
4. **Memory Management**: Windows has both physical and virtual memory. Processes use a virtual address and then are converted into a physical memory address in what Windows calls Blocks and Pages. These two technologies work in unison to offer reliable memory management in Windows.
5. **Distributed Systems and Networks**: Popular development engines such as Unity, Unreal Engine, and Godot offer built in tools that handle the communication for data between devices and operating systems. The game will heavily rely on a server, so a stable connection is vital.
6. **Security**: The best way would be to ensure all data is encrypted. AES256 is a widely known encryption algorithm. Additionally, sensitive information such as an email and password should be encrypted using a hash algorithm. So that if a breach occurs on the system the data would be secure since only the user would know the information.