

Draw It or Lose It

# **CS230 Project 1**

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 | 09/14/22 | Christopher Banner | <Brief description of changes in this revision> |

**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 looking to create and develop a web-based game that is accessible and executable across all platforms. “Draw It or Lose It” is the title of the game they want to create and currently Android is the only compatible platform. The goal of the game is for several teams, consisting of four rounds that are one minute in length. A team selects a picture from the library and a team then makes guesses until the one-minute timer runs out. If the team is unable to answer properly, the other teams have fifteen seconds to steal the answer.

## [Design Constraints](#_2et92p0)

* Requires more then one team to play
* Each team requires more than one person
* Must create unique team names and allow teams to check if the name is already in use
* Can only play one game at a time
* Game needs to run on all platforms

The create must follow these constraints when coding and designing this game. These constraints only apply to the game software and the application development still needs to be looked at. The company has put an emphasis on this game running on all platforms, as it currently only runs on Android. To continue, we need to get the software to function on all mobile devices, as well as all desktop operating systems. This will require either a code re-write into another language, or the ability for the code to inherit other languages.

## [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 entity is there to create a relationship between Game, Player, and Team classes. What this means is, the classes will all inherit information from Entity. Inheritance is the way that we display this within the UML diagram. This inheritance means that the classes will share common information that is important to all of them making the code efficient by using a superclass. When using a UML diagram, we define a “Has-A” relationship with the term aggregation. Examples of this would be Team and Player classes. This means that a particular class has a reference to an instance in another class. GameService shows that it references Game, Game has a reference to Team, and Team has a reference to Player.

**"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** | The Mac has flexible terminal commands that help to configure server, access, and make changes.  **Advantage**  Upgradeable and includes several options for web hosting.  **Disadvantages**  This OS is not preferred when it comes to web hosting. | Similar to Mac but more cost effective.  **Characteristics**  Secured is most preferred  **Advantages**  Linux includes a security system that catches flaws before they create issues. This makes it the most preferred for web hosting.  **Disadvantages**  Applications are difficult to find for web hosting. | Has the most software to chose from when compared to other operating systems.  **Characteristics**  Dominates other platforms and it considered a close platform.  **Advantages**  Lower loading times, plethora of resources, and high comfort level for most users.  **Disadvantages**  Highly susceptible to viruses and hacking attempts. | Specifications are much better in other platforms.  **Characteristics**  Hugely popular due to its mobility.  **Advantages**  Cost effective and have a larger market of users  **Disadvantages**  Mobile devices have poor security making the server difficult to maintain |
| **Client Side** | The level of expertise is moderate, and the cost is fairly comparable to Windows. Expresses the requirements and constraints for the development process so that the application is compatible across all platforms and mobile devices. | Higher level of expertise required along with minimum cost. Expresses the requirements and constraints for the development process so that the application is compatible across all platforms and mobile devices. | Lowest level of expertise and time required and similar in cost to the Mac OS. Expresses the requirements and constraints for the development process so that the application is compatible across all platforms and mobile devices. | Allows flexibility for the client, as well as the developer so they can view updates from any location. Security issues make it more difficult to implement. |
| **Development Tools** | Swift is the most common when trying to run languages on Mac. Mac can run languages like CSS, HTML, JavaScript, as well as others, while supporting libraries to support other general-purpose languages like Java or Python. | Linux can work with several IDE’s, which makes it an easy tool to use. Linux can run languages like CSS, HTML, JavaScript, as well as others, while supporting libraries to support other general-purpose languages like Java or Python. | More user friendly than Linux and is compatible with the same software and applications. Microsoft can run languages like CSS, HTML, JavaScript, as well as others, while supporting libraries to support other general-purpose languages like Java or Python. | Has the ability to create an unlimited number of applications using Swift or Android. Both languages and software can be run. Mobile devices can run languages like CSS, HTML, JavaScript, as well as others, while supporting libraries to support other general-purpose languages like Java or Python. |

## 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 would be the best place to start, as it is the most user friendly and has the largest selection of software. All IDE’s can be run on Windows, so you will not run into limitations in that department.
2. **Operating Systems Architectures**: Windows includes services that all Windows-based applications use and enable applications to show a GUI or graphical user interface when accessing resources. User accounts allow you to use these services that include graphics and multimedia, messages, and web services.
3. **Storage Management**: Windows 11, much like Windows 10, includes a storage sense feature that allows you to manage your files on your hard drive and see how much space the files are using. You can choose the storage location of files and apps that make them easily accessible. The cloud is also available for storage and Windows includes a built-in storage system that makes is simple to create and move projects.
4. **Memory Management**: The game will include a library that holds all the images required for the game. Windows storage options will allow you to easily move the pictures outside of the default storage folder. This allows you to work from one secure location that will include you IDE files that is easily accessible.
5. **Distributed Systems and Networks**: The operating systems are all different, so several publishing software’s have been considered. Develop 4 enables cross-platform game creation making it an ideal IDE that can be run on all devices. This will enable you to export the game file to any platform and will allow the ability to cross-play. The company will need to have strong enough servers to prevent outages and connectivity issues. These servers will need to have the strength to support large player volumes and include backup power to accommodate any power outages.
6. **Security**: Windows includes a security protection software that helps with the security. Along with this built-in service, a secondary service is recommended for securing user data and information. Windows also scans for malware and virus on a regular basis that helps mitigate security threats. Automatic updates help to attack and defend against the ever-changing security threats to keep the system and the user safe from these attacks.