

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 | 11/17/22 | Michael Antoniazzi | First Draft of Document |

**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 approached us with the task of designing and implementing a web-based game that can run on a variety of platforms. The goal of the game will be to have multiple teams consisting of several members guess an image randomly pulled from a library of images as the image renders. The team will be able to make guesses of the image for up to one minute and the game will have four separate rounds of play. If the image is not guessed correctly in the time limit, the other teams will have a 15 second opportunity to guess the image for themselves.

## [Design Constraints](#_2et92p0)

* Need for multiple teams consisting of multiple people with simultaneous access to the game
* A method for distinguishing the various teams have unique names or identifiers
* Game must run on multiple platforms
* Only one instance of the game can exist at any given time by having unique identifier for the instance

## [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 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.**

Entity class creates the relationship between Game, Team, and Player classes. All these classes will inherit from the Entity class. Each class will be able to share references like “name” and “id” thanks to this inheritance which will make Entity the superclass of this program. We can see in this UML diagram that the game makes use of both Inheritance and Polymorphism through the Entity class by overloading the constructors. Team and Player classes also use Aggregation which references other classes and can best be referred to as a “has-a” relationship.

## [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** | Easy and flexible command in the terminal to make changes to the server or alter its configuration.  Advantages:   * Upgradeable * Multiple options for web hosting options   Disadvantages:   * Less preferred for web hosting services | Similar to Mac in regard to the flexibility and ease of use.  Advantages:   * Most preferred choice for web hosting services * Security flaws can be caught before they become an issue   Disadvantages:   * Difficult to find applications that support the web hosting needs | Most software available when comparing to other Operating systems.  Advantages:   * Less loading time * High resource requirements   Disadvantages:   * High risk of viruses * Poor tech support | It is generally better when the sever is immobile and can be accessed in a single place.  Advantages:   * Cost effective * Wide reach * Higher rates of compatibility   Disadvantages:   * Poor security |
| **Client Side** | A mid-level amount of time and expertise are necessary. Costs are comparable to that of using Windows. | High level expertise is required to work with Linux. Cost is very low. | Low level amount of expertise is required to work with Windows. Costs are comparable to that of using Mac. | Lots of flexibility provided to clients and developers but can be difficult to implement. |
| **Development Tools** | Swift is the most widely used language on the Mac OS. You can also work with languages like HTML, JavaScript, CSS and support for Java, Python, Ruby.  Tools like Notepad++ can be used. | Linux supports a broad range of languages including C/C++, Java, Python, JavaScript, and Swift. Tools and IDE’s include Visual Studio, Eclipse, as well as Notepad++. | Windows supports a broad range of languages including C/C++, Java, Python, JavaScript, PHP, Red-Lang, Go, Object Pascal and Swift. Tools and IDE’s include Visual Studio, Eclipse, as well as Notepad++. | Mobile Programing languages include both Android and Swift for application development. |

## 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 would recommend Windows as the OS to use due to its wide use and lower required expertise. Many different IDEs and languages are compatible with Windows and costs should remain acceptable with this platform.
2. **Operating Systems Architectures**: Windows offers a wide range of services like Graphics and Multimedia, messaging, as well as web services so I feel it is the most ideal to handle the tasks of the game service.
3. **Storage Management**: Windows continues to offer easy and intuitive storage management software through updates to the Windows OS. Windows 11 includes many ways to manage storage devices, including the Disks & volumes settings and Disk Management.
4. **Memory Management**: This game will require a large library of images which will require excellent memory management in order to maintain good performance.
5. **Distributed Systems and Networks**: Strong servers with high-speed connections will be required in order to ensure a large user base can connect to the multiple game instances without any drops in performance. Flutter is an IDE developed by Google that can be used to develop cross-platform applications. It allows you to build high performance apps using one codebase, is open source, natively compiles, and has automated testing features.
6. **Security**: While Windows does have some great security features included, I believe it would be best to find an even better solution to security to prevent any bad actors from obtaining any user data or information.