

<Draw It or Lose It>

# **CS 230 Project Software Design Template**

Version 2.0

## 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 | <05/17/22> | <Perry Matthieu> | <New outline for the development of Draw It or Lose It.> |
| 2.0 | <05/18/22> | <Perry Matthieu> | <Added 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 needs a game in which multiple players on one or more teams compete to guess a gradually generated image. The first team will have 30 seconds to guess the Image during which time the image will gradually appear and be fully visible at the end of the timer. Every other team will have 15 seconds to decide on a single guess if the first team is unable to come up with an answer. Each player and team must have a unique name.>

## [Design Constraints](#_2et92p0)

<As the game has the possibility of single team games, The game may need different rules to accommodate for single team games.

The game must run on multiple platforms. This means code will need to be written In various programming 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)

<As described by the model below there may be 0 or more games per game service, 0 or more teams per game, and there may be 0 or more players per team. The Game, Team, and player classes all inherit the entity class. The SingletonTester class is used by the ProgramDriver 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)

Server side:

* Does each of the operating platforms offer a server-based deployment method where the website will be hosted? YES
* What are the potential licensing costs to the client, The Gaming Room, for the server operating system? If Linux is used it is free of cost as it is open source.

Windows is another decent option. Windows servers are slightly more secure and there is more support available, however, Linux is still pretty secure and is significantly less costly. I would recommend Linux.

Client side:

What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices?

-One thing to take into consideration when developing software for multiple platforms is the UI. With desktops you have plenty of screen space to work with however on mobile devices the available screen space is significantly smaller and therefore the user will likely need data broken up into sections to be navigated to.

* Windows

Development Tool/IDE - Visual Studio

Programming Language - Visual Basic (default)

* Linux

Development Tool/IDE – Many but eclipse is the most popular

Programming Language - C is the default

* Mac

Development Tool/IDE -XCode

Programming Language - Swift (or Objective C)

* iOS

Development Tool/IDE -XCode

Programming Language – Swift

* Android

Development Tool/IDE - Android Studio (default)

Programming Language - Java

* Web Application

Development Tool/IDE- Visual Studio

Programming Languages- HTML,CSS,JAVASCRIPT

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** | < Flexible terminal commands to configure  the server, access, or make changes.> | <Cost effective comparitavely. Can run on any hardware> | <More beginner-friendly but more prone to cyber threats > | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> |
| **Client Side** | <Moderate expertise and time required.  Cost similar to windows> | <Maximum expertise and time required.  Minimum cost.> | <Minimum expertise and time required.  Cost similar to mac.> | <Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices.> |
| **Development Tools** | <Most development tools are accessible on mac, linux, windows and mobile devices, however, Objective-C is popular on mac.  (ex of dev tools usable: HTML, CSS, JavaScript, Java, Python, PHP, and Ruby.> | <Majority of development tools such as: HTML, CSS, JavaScript, Java, Python, PHP, Perl, vala, Ruby, etc.> | <Utilizes about the same tools as Linux> | <can utilize most software that windows and Linux utilize, however, swift and java are the most used because they are primarily used for android and iOS app 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**: <Windows would likely be the best choice. It’s minimal necessity of expertise will make it widely accessible. Windows also has a wide variety of available IDE’s to work with.>
2. **Operating Systems Architectures**: <Windows provides services used by all Windows-based applications that enable applications to show a GUI while accessing system resources. Windows command prompt power shell utilization offers quick and easy server configuration settings.>
3. **Storage Management**: <Windows allows specification of storage locations of apps. Windows OS allows cloud storage as well. Windows storage sense removes files that aren’t needed to clear up drive space. Windows file management system is very user-friendly and intuitive so large projects can easily be maintained. Cloud storage may be a good option as costs and resources will be more effectively managed with game expansion in the event that that occurs.>
4. **Memory Management**: <Memory management in windows includes physical and virtual address space.>
5. **Distributed Systems and Networks**: <Distributed systems and networks may run into issues such as routing and congestion problems. Though these issues can happen, these systems offer easy communication and coordination between one another. Some other common issues of using a distributed system are absence of a global clock, lagging computing performance, and connection problems on client-side due to simultaneous computations>
6. **Security**: <Windows has built-in anti-spyware software as well as account control settings. It also ensures that authorized operating platform modifications are not made without acceptance from the administrative user(s). While this software is a bonus It likely will not be sufficient. A VPN may help to prevent cyber-attacks that occur during the client-server data transfer process but will likely be only a small addition to security. I would recommend a different mode of security than windows built-in security and/or a VPN as they are not all-encompassing.>