

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 | 05/17/2022 | Sarah Snow | Initial Design Template |

**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)

Creative Technology Solutions would like to create game software that is web-based, serving multiple platforms. The software should allow for four rounds lasting one minute each. The concept of the game is to guess the puzzle being rendered by the computer within a time constraint. The software should allow for one or more teams with multiple players assigned to it with unique names, and only one instance of the game should be existing in memory at any given time.

## [Design Constraints](#_2et92p0)

This game is going to be designed to serve multiple operating platforms, which increases cost and time needed to create the software. The hardware needs for testing will require multiple kinds of hardware, given the different platforms to be served by the software, so there will be an increase in cost and time as well.

## [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 com.gamingroom package contains multiple classes that create the program. The ProgramDriver class contains the main method that drives the entire program, and uses the SingletonTester class to test whether or not there is a single instance of the game going at any time. The Game, Team, and Player classes all inherit from the Entity class, which means they have their own unique attributes but also use the attributes within the Entity class. The GameService class does not inherit from any class, but uses the information from the Entity class and its subclasses to then create the game.

**"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 is popular for web-hosting and in general is upgradeable.  A disadvantage is even though it is popular for web-hosting, it is actually less preferred. | Linux is cost-friendly and secure. A disadvantage is that it is less popular, which makes finding applications to more difficult. | Windows is popular and there is an abundance of software available. It requires less time to load and is comfortable to use. Disadvantage is that it is less secure. | Popular, and has a wide demographic, given that almost everyone has a mobile device. Disadvantage would be that it is technically less secure. |
| **Client Side** | Requires a moderate amount of expertise, cost, and time. | Requires a maximum amount of time and expertise but is less demanding in cost. | Minimum expertise and time needed, with a moderate cost. | Requires a moderate amount of expertise, time, and cost, but allows for updates more easily. |
| **Development Tools** | For running on a Mac, some relevant programming languages and IDEs include Swift, HTML, CSS, JavaScript, Python, Ruby. s | For running on Linux, the languages and IDEs that are relevant would be Eclipse, Visual Studio, HTML, CSS, JavaScript, Java, Python, Ruby. | Same as Linux. | Relevant languages and IDEs for mobile devices would include Swift, Visual Studio, Eclipse, HTML, CSS, JavaScript, Java, Python, Ruby. |

## 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. It requires the least amount of expertise and time, with a moderate cost.
2. **Operating Systems Architectures**: Windows has services that all Windows-based applications can use, giving it a lot of support and features.
3. **Storage Management**: Windows provides solid storage management with the ability to use cloud storage as well, so there is less likely to be any unintentional deletion, and the ability to back up data.
4. **Memory Management**: The application will require a large database to hold the stock pictures from which the players will guess. Windows will allow for a large file and to have the whole project stored together for easy access.
5. **Distributed Systems and Networks**: To contend with outages or connectivity, the servers used will need to have the strength and ability to support larger volumes of players, as well as backup power. There are IDEs available to develop applications to be used on multiple platforms which can help with dependence.
6. **Security**: Windows offers security software that is built into it. It offers real-time malware scans and automatic system updates to contend with any security threats.