

The Gaming Room: Draw it or lose it

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/24/2024 | Landon Alexander | * Summarized the task given to us by our client * Briefly make a summary of the requirements needed for the project. * Detailed the design constraints given by the projects demands, and explain why these constraints are considered important * Explained the domain model used to operate the game. Explained how the new class Entity contrasts with its other classes (Game Service, Team, Player, Game). Also explained how program driver and singleton tester are needed for the software to work when using the other classes given. |
| 1.1 | 6/6/2024 | Landon Alexander | * Added in details about each service provided (Mac, Windows, Linux, Mobile Devices) * Explained each of the services development requirements (Server Side, Client Side, Development Tools) |
| 1.2 | 6/20/2024 | Landon Alexander | * Added characteristics of specific software used for “Draw It or Lose It”, and techniques specific to various systems architectures |

**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)

This project requires a web-based game to be produced on software that can develop a game for multiple platforms. The game in question is inspired by the android game “Draw It or Lose It “, which is currently only available on Android stores. The software must be made to implement the game on multiple devices, while also being able to transfer android coding to retain the original’s vision.

## Requirements

The gaming room does not know how to make the environment, so we must develop the game from the ground up. While also having inspiration from their previous game on Android. The game must also be able to retain its vision while being transferred on multiple platforms. The game has mechanics that need to be verified during the development process.

* A game can have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

The game also needs developers who are pros inside of the gaming industry as this game requires many system requirements for it to work. Both an Android and console game developer is needed as we need to understand what is consistent when programming android games compared to console games with these mechanics.

## [Design Constraints](#_2et92p0)

Design constraint one (Retain its vision) is important as transferring android code into multiple platforms could jeopardize the vision of both the game and its mechanics. Having consistent communication during its development is key to keeping the mechanics the same while making quality of life changes to its operations.

Design constraint two (hire developers who are professionals in android and console) is important because having a leadership that has knowledge of these platforms can lead to the project having a better view on how it should operate. With great leadership the project can be developed faster as everyone is on board due to the director understanding the fundamentals of the software they use.

Design constraint three (Understanding what programming is needed for the game’s development) is important for the project to be a success. The mechanics must be implemented smoothly without any coding hiccups. Mechanics demand a development process that is coordinated and responsive towards team members. Once the 4 requirements of the task are completed, then we can focus on polishing the games details and quality of life, otherwise the mechanics come first.

[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 base class is created to hold the id, name, and string for all three of its subclasses (Game, team, and player). The player class extends entity to hold information of each player who enters the game, and then outputs that information with the overwritten method. Each of the subclasses has a relation to one another, for the player-team relation, it holds together the requirement of having “a way for multiple players to be on one team.” Game-team relations hold a private list of teams and verify that each team name is unique (with the same method used in the player class). This implementation solves one of the core mechanics “the game will have the ability to have one or more teams involved and must each have unique names that can be checked.” Game Service-game relations are also key to the code working as the game service class holds a private list of games and an iterator to check if the game has a unique name. The class is a singleton class meaning that only one instance of the game can exist in memory. This is operated by a Singleton Tester class and a Program Driver class that harbors a test singleton method that, when succeeding, will conclude the data sent to it to make the game function efficiently.

**"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 must 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 can create simple server-based development and provide great security against attacks. It also has amazing technical support with the products given on the system.  However, it has limited hardware options, and it requires mac systems that use macOS. Also, it is expensive software compared to the other options. | Linux is an open-source software that can be used even on the lowest hardware. It is also cost-effective compared to the other software options. Linux is open-sourced, meaning it can be easily modified and updated to your own preferences.  However, Linux is known for being a program with not a lot of software to use, unlike windows. | Windows is the most easily accessible program from the bunch, and it also gives a long range of software support options.  The program has the fastest update schedules out of the other programs due to Microsoft making sure it is at efficient processing.  However, it has less secure network compared to Linux and Mac. It also updates only with Microsoft, and some updates can detriment the program. | Mobile Devices can hide code from the user, and store data that can be easily operated. It also can optimize queries for server-side databases through calls.  However, for it to interact with server-side it must go through cloud services, or physical servers.  It is also expensive to maintain as it needs regular updates for users to access the program without error. |
| **Client Side** | Macs have a wide range of supported web browsers that offer many tools for developers. Development time on the Mac is cut by half due to its performance focusing on speed and instant response time.  However, even with the benefits it only applies to apple products like the MacOS. | Due to the service being open sourced, it harbors a broad range of supported browsers harboring tools for developers.  Linux is fast in the development and deployment of products.  The service being open sources also gives it every web browser access because of developers tinkering with it to make it better. | Windows harbors a vast range of web browser support that inhabit tools useful for developers.  Windows has amazing cross platform testing due to the program harboring code that can be universal for any changes.  However, it is difficult to test MacOS browser as it does not harbor universal code for that program. | Mobile devices are simply the best in android app development, along with apple app development.  However, it is extremely difficult to test on other browsers and environments that are not mobile devices.  It will take longer for development to happen due to the need to create multiple versions for different operating systems and devices. |
| **Development Tools** | The mac system offers documentation for deployment on MacOS, which is software that is faster and more proficient in workload. It also offers Eclipse/java in its programming. Virtual machines can be used on the Mac, meaning that Windows and Linux can be ran on the system | Linux can be developed at any time due to the software being well versed in flexibility.  Services like Windows and MacOS can be easily run with virtual machines.  The software being open-sourced gives it an edge in development as the community is vast in creativity and upgrades. | Windows has a plethora of tools that can be used for development.  Well known tools ranging from Visual Studios (JavaScript, C++, C#, HTML), and Eclipse (Java).  Windows also has documentation on windows deployment, along with the freedom to be opened at any time. | Mobile Devices offer good services for development that can be helpful for android and apple apps.  Known software used ranges from Android Studios, Eclipse, Xcode, and Swift. |

## 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**: Out of all the operating systems given, Windows is the best for “Draw It or Lose it” The reason for this is because it integrates with android well with its options like Xamarin, react, and android studio. It also helps developers design and develop apps for Window PCs using Microsoft’s products. Windows harbors the largest user base and operating platform for game development thanks to many users implementing features benefiting developers.
2. **Operating Systems Architectures**:

Windows can separate the operating system from user mode to kernel mode whenever it opens an application. The user mode is less privileged than the kernel system and you are not allowed to access the system resources directly. However, most of the code work is managed with this mode since it has protection unlike its counterpart. For users to gain access to certain resources, it must go through kernel mode. Kernel mode is the core program used for everything as it is a master mode in the system. This means it has freedom to access hardware components, memory management, and networking. However, this system is reserved for the lowest level, and any crash in this system can completely hinder the entire PC.

1. **Storage Management**:
2. The storage management in windows will be managed by IBM Cloud. This is because it is easy to learn, and it has great documentation for developers to research during development. Another recommendation is Microsoft Azure as it has good customer support and a consistent update service to make sure there are little errors.
3. **Memory Management**:
4. Windows 11 is the latest version of windows, and so far, it performs solid in memory management. Thanks to the advanced algorithms for memory allocation, windows 11 makes sure the system resources are used more effectively. The new OS is built primarily to bring better performance by relying on the RAM and CPU more effectively, while expanding the RAM if the physical memory is taxing. It also utilizes virtual memory space that stores the data into a table each time it processes, and it can be used if allowed to share.
5. **Distributed Systems and Networks**: I recommend IBD cloud as the cloud service provider since it provides great flexibility and scalability with its cloud workloads. It is very cost-effective in network management, which will be needed in creating multiple teams with multiple players in many games. IBM also has a cloud content delivery network that can minimize latency and eliminate risks in facing network traffic jams. For a network-based game like “Draw It or Lose It" this is phenomenal as it helps negate any network issues that could plague the game post launch.
6. **Security**: IBM cloud is recommended again due to its immense focus on security support. It can be managed quickly towards advanced threats all the time thanks to its cloud security monitoring in real time. Its cloud security uses threat intelligence to understand threats that can camo their way into the system, like ransomware, or malware. IBM uses AI technology to detect security incidents using machine learning algorithms that collect data on the most effective hunting strategies for certain threats. Thanks to IBM’s flexibility in its design, it can design security solutions based on the needs of organizations. Since we need security for our game’s development, our problem would be considered mid-sized difficulty, meaning that it would provide flexible deployment options to tailor to our security requirements.