

“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 | 01/23/2022 | Kyle Fisher | Summary, Constraints, UML Class diagram, Requirements and Recommendations. Updates to OS Evaluations |

**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 is requesting development of a multiplatform game software. The initial development of the game “Draw It or Lose It,” requires the following:

* A game will have the ability to 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.

We will create a Java program with classes to handle player, team, and game objects that inherit from an Entity superclass. There will be a service class that checks for existing games to make sure only one game exists with a specific id within memory at a given time.

## [Design Constraints](#_2et92p0)

Operating System constraints:

* Which operating system will the game be used on?
* Moving from Android app to multiple OS platforms

Connection constraint: (Needed for involving multiple OS)

* Is there a minimum connection speed needed to run the software?

Schedule and Budget constraints:

* To fit within timeline, a schedule needs to be worked with client.
* A budget will also need to be requested from client and worked.

## [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 superclass “Entity” creates an object in Java using private variables “id” and “name.” The superclass creates methods to return variables of the three classes that inherit from it, “Game,” “Team,” and “Player.” The Game class creates a private list of teams, adds teams to the current list in the “id and name” format, and has a one-to-many relationship with the Team class (which means one Game can have many Teams). The Team class creates a private list of players, adds players to the current list in the “id and name” format and has a one-to-many relationship with the player class (which means one Team can have many Players). The Player class creates a Player object in the “id and name” format. The GameService class creates a list of games, adds Game instances to the list and holds the next Game, Player, and Team ids for assignment. The GameService class creates a singleton pattern meaning that only one instance of a Game exists with a unique id in memory at a given time. This keeps from overwriting saved game data. The GameService class has methods to return Game names, id’s, the amount of Game’s in memory, Player and Team id’s and has a one-to-many relationship with the Game class (which means one GameService can have many Games). The ProgramDriver class uses the SingletonTester class to test the behavior of the singleton pattern created by the GameService 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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Flexible terminal commands to configure the server, make changes and access.  Closed source OS.  Any updates or tweaks will be solely released by Mac.  Any update to the software will require payment from the web host to obtain the new version.  Offers OS-X Server for web hosting with 2 different types of licensing: 10 Client - $499  and  Unlimited client - $999.  Client access limit is only for file sharing. Offers substantial discount upgrade package with new OS versions using “Apple Maintenance Program” | Terminal access the same as Mac, but more cost and user friendly.  Free software that anyone can change, therefore webhosting is often cheaper.  No licensing fees.  Works on over 70% of webservers. | Not open source as Linux is, however, contains more available software programs.  Closed source OS.  Any updates or tweaks will be solely released by Windows.  Any update to the software will require payment from the web host to obtain the new version.  Offers leasing or owning server licensing options.  Leasing is between $20- 125/month.  Purchasing a standard license is $975 and a data center license is $6155. (Data center offers multiple virtual machines to be licensed under one server. | Specifications are better in immobile devices. OS are closed source.  Backend server hosting using a Third party mBaaS. (Mobile backend as a service).  Provides access to features like search, data storage and authentication.  Some hosting services are free. Offering up to 10GB on multiple websites.  Or pay as you go services for an average of $0.026 /GB |
| **Client Side** | Moderate time and expertise required. The cost is like Windows OS.  Requires highest-end hardware to run programming.  Cloud application support through MacInCloud. | Maximum Expertise and time required for support. Minimum cost compared to other OS.  Typically, good for startup companies considering its open source. | Minimum Expertise and time required for support. The cost like MacOS.  Requires high-end hardware to run programming, typically not as expensive as Mac, yet more expensive than Linux  Cloud support through Microsoft Azure | Provides flexibility to clients and developers to view updates while mobile. More difficult to implement software programs. |
| **Development Tools** | Main language: Swift and Objective-C  Common languages: Any language for which an interpreter exists for Mac. i.e., Java, Ruby, Python. Although Windows based languages such as C, C#, and C++, interpreters seem to have trouble implementing cross platform.  Eclipse, PyCharm, and XCode IDEs are powerful on MacOS | Main Languages:  Python and C  Common languages:  Any Language for which an interpreter exists for Linux. Which is basically every language because of the open-source format of Linux OS.  Java, Perl, PHP, C, Python, Google Go, can all be used with Eclipse IDE. | Main Languages:  Visual Basic and C  Common languages: Any Language for which an interpreter exists for Windows.  Visual Studio and Eclipse are powerful IDEs for windows.  Eclipse supports languages such as:  Java  C  C++  Perl  Python  Ruby and more | Mobile devices with android OS follow a close format to Windows OS. Apple iPhone obviously follows Mac OS languages and formats. Although mobile applications can be developed through Amazon Web Services which offers a broad set of tools for Android applications. |

## 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 recommend using Windows Operating System and developing the application in C++ language using Eclipse IDE. I personally use Mac OSX, but Windows will allow for more tools and the Lightweight Directory Access Protocol (LDAP) supports C and C++ languages in the client/server model. The function of LDAP is to enable access to an existing directory.
2. **Operating Systems Architectures**: Windows is a layered design containing two main components, user mode and kernel mode. Windows architecture is comprised of HAL, driver, microkernel, and executive services. Kernel mode is often the most trusted function of the operating system and has complete and unrestricted access to the underlying hardware. Code running in user mode must delegate to system APIs to access hardware or memory which allows kernel crashes in user mode to almost always be recoverable and protected.
3. **Storage Management**: There are many ways to provide storage for the Draw It or Lose It software. Although, considering the addition of image packages to add on to the original library of images for game play, I would suggest using a cloud-based storage management system such as Microsoft Azure. Cloud storage allows for unlimited expandability, and you’ll only pay for what you use. Because of the simplicity of the game, the engine execution will be run with little to no latency, and the cost of storing on the cloud will be cheaper than purchasing Windows servers and licenses and continuing to add on while managing the infrastructure of the back end. Also, local hardware systems can be easily manipulated and damaged, while cloud servers are much more protected and secure.
4. **Memory Management**: Newer versions of Windows OS use a function called memory compression which is designed to improve the responsiveness of the operating system during heavy use. When the amount of RAM available in the system reaches a certain percentage, the OS begins compressing “pages” or blocks of memory so that they take up less space. After compression the memory manager moves them to another part of memory called the compression store. When the data needs to be accessed, the manager locates and decompresses that page. Since the data never leaves memory and is never written to the hard disk for temporary storage, swapping is made much faster and more efficient.
5. **Distributed Systems and Networks**: A distributed system consists of a web application or distributed application on a multitude of networked computers that will run the client side of the task. These applications will split the task into pieces and deliver the data to a central server(s) for data storage and execution. In this case, the application will be housed on a cloud-based platform and interacted with on a web page connected via internet; handing off workloads to dozens of cloud-based virtual server instances that are created as needed and terminated once the task is complete. Although, the cloud-based application can be accessed via internet from any OS via front-end software formatted for that specific OS. The only large downside to cloud-based storage is internet connectivity and its volatility. If connection is disrupted, be it a local storm or a system failure, access to the cloud storage will be disrupted.
6. **Security**: Windows OS uses many different forms of user information security when operating locally and connecting to cloud-based services. Staring with a “secure boot” to help prevent malware and corrupted components from loading when the device is starting, to advanced cryptography and certificate management. Cryptography and encryption use code to convert data so that only a specific recipient can read the information by decrypting the code with a “key.” This security feature is extremely important when connected to an internet program and transferring data and personal information to cloud-based servers. Windows system also safeguards at-rest data using encryption to protect against virus attack through malicious applications. Windows application has firewall capabilities that can be modifies to allow and prevent access by network traffic to your device. It also supports Internet Protocol security (IPsec) which will require authentication from any device attempting to connect or communicate with your device.