

Draw It or Lose It

# **CS 230 Project Software Design**

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> | Alexandrea Teigeler | The Executive Summary was defined, the design constraints were listed and given context. The UML Design Document was explained, and the OOP principles were outlined on how they were used with this application and UML. |
| 1.1 | <02/06/2022> | Alexandrea Teigeler | The Evaluation of Operating Systems characteristics, advantages and weaknesses for hosting web-based software, and the tools that are used to develop for them were outlined. |
| 1.2 | <02/20/2022> | Alexandrea Teigeler | The final recommendations for The Gaming Room of their application “Draw It or Lose It” is outlined. The benefits of the recommendations are explained. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming room has approached CTS to set up a web-based environment for their game Draw It or Lose It on multiple platforms. This game will allow teams to compete against another in timed rounds. Each team needs a unique name and identification number, as well as each player needs their own unique name and identification number.

## [Design Constraints](#_2et92p0)

* Needs to be designed for multiple platforms.
* Budget considerations.
* Current Application architecture and Framework.
* Only one instance of a game can exist at a time.
* Unique identifies for each game, team, and players within each team.
* Programming Languages being used and the corresponding developer tools.
* Timeline on the project on what can be achieved and deliverable.
* Internet Connection and Cloud usage.

## [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)

OOP (Object-Oriented Programming) principles are used as the basis and foundation of creating the UML Diagram and designing a program that is efficient and effective. Using these principles, troubleshooting a problem is streamlined and creating concise classes that build upon another. Within this UML Diagram for Draw It or Lose It, the Game, Team, and Player class all inherit from the Entity Class. This inheritance is purposeful, giving each class similar traits yet allowing them to have different values and attributes depending on how the class chooses to use them. Encapsulation of the private values protect this data from being accessed and changed. Polymorphism is achieved with the use of the super class and constructors that overload the methods depending on the types of data being called. This is show in the GameService class with the methods getGame. There are two methods with this name, but as each one calls for different type, they will only be performed depending on the type of information being sent as it corresponds to these parameters.

**"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** | Currently, most newer OSX servers cannot be reasonably used for creating servers. However, if used, they work most notably for Mac-Only shops. | Linux is a popular OS because of what it inherits from UNIX. It is stable and efficient, and server hardware can run for a long period of time without a need for rebooting. It has low configuration requirements, which means it can run on older hardware, and has strong security since the community will quickly find vulnerabilities and release patches for other users.  Using Linux for web hosting offers better security, is compatible with most web hosting software and hardware. Linux is open source and can be used and modified freely. However, the use of Linux requires more expertise, working with the command line and must manually edit and configure files. | Windows is the most popular OS and offers vendor support after sales. It uses a windows server as the OS and offers window specific technology not available for other operating systems.  Using Windows for web hosting can be easier to use and can develop window-based applications. However, it can not handle high volumes of activity. The servers require frequent rebooting to run as intended, and costs more to use than other servers. | Mobile application website hosting is becoming a popular choice for businesses. It is pay for what you need and offers scalability, which is good if there are seasonal fluctuations in users. There is an improved speed in using the cloud, as data transfers are typically faster and offer a robust back-up scheme.  Depending on the business used for cloud hosting, the security has advantages and disadvantages. The data is accessible from any device with an internet connection, which can pose significant threats. It would be important to find a service that offers encryption for data. Google Cloud web hosting as an example, offers data confidentiality, uses role-based authorization, follows cloud compliance laws, and has Audit rolls for administration usage. An important aspect of using cloud hosting is data management and having a strategy of how to implement it, such as Machine learning tools to track how it is being used, or data visualization tools. This is necessary in order to reap the full benefits of the ‘pay for what you need’ idea of using the cloud and tracking seasonal usage. |
| **Client Side** | Costs:   * $99/yearly Developer fee * Developers receive 70% revenue * Customers tend to pay for applications   It can be difficult to be approved to enter the Apple Store, however, the review team offers good feedback on what the application needs to be accepted. Once accepted, there is a potential to make 2 times the amount from downloads as compared to the Google play store. It is important to utilize and understand how their search works, which requires attaching keywords to the application in order for the user to search and find it. There are developer resources available to help with marketing, promotion, application launch, and monetization. | Developing and selling a game for Linux, most of the users expect the software to be free, or open source. More often, most Linux users will use window emulators to play games. | Costs:   * Requires a onetime $25 Developer fee. * Developers receive 70% revenue * Customers often prefer free applications   It can be cheaper initially to launch a game through the Google play store, and the process to submit and be accepted is not as strict. Google play store records at 2 times the number of downloads compared to others. However, this means the market is saturated with products and is harder to stand out. There are developer resources available to help with marketing, promotion, application launch, and monetization. | Using Native applications as opposed to a web-based application offers a faster application and is safer for the user. However, it requires consistent maintenance which requires time and money. It also requires approval from the stores being used to launch the application. |
| **Development Tools** | Apple has specific web developer tools for both macOS and IOS on their search engine safari: Web inspector. It uses CSS.  Xcode is used for Mac software development, using swift and Java languages.  Other tools and API’s available: Xcode Cloud, Xcode 13, camera APIs, StoreKit 2. | Linux uses C/C++ compilers for the majority of their software development. Sublime text offers a variety of development tools for software or website use and has a plugin manager to control functionality.  Seamonkey Composer offers the ability to create websites and work with HTML and CSS. | Eclipse is used for Java development and website design. Visual Studio offers usability for several different languages and performs well for several platforms and is the main windows software developer tool. Most languages can be utilized for window software games and applications, but C++ and Java is a common language. | Xcode is used for iPhone development, which can be developed using the swift and Java languages.  Kotlin and java is used for Android development, using the Android Studio developer tools IDE. |

## Recommendations

* It is recommended to use a Windows operating platform for Draw It or Lose It in order to expand for other environments. Windows development offers the widest variety of tools and offers vendor support after sale.
* Development with Visual Studio for will offer a strong IDE that can be used with multiple languages, a must when developing for several systems and deploying these systems meant to communicate amongst each other. Along with using this OS for its development tools and access, the ability to develop a server and website is better suited than others such as the Mac. While Linux offers better security and continual running without interrupts within the server side, it is recommended that for the most optimal storage and memory management, using a cloud service would allow The Gaming Room to scale as needed without the overage charges of purchasing equipment that isn’t being utilized.
* It is recommended that The Gaming Room use a Layered Architectural Pattern.
  + The system is layered into several subsystems and tasks that they perform and responsibilities they alone solve.
  + The change to each layer does not affect the other, such as changes to the game itself would not affect the current user session, and the change in network wouldn’t affect the current deployment of the application.
  + As changes made for a specific platform, if changes are not needed or ready for deployment on another, the change would only be deployed for that specific platform, making this pattern suitable for this problem.
* Development using a REST style API will enable the application to communication across various platforms using the standard and accepted code setup that manages and examines requests from the client.
  + It dramatically accelerates productivity and offers a large, prebuilt library to draw from.
  + When deploying for multiple platforms, REST API enables a uniform interface which simplifies the overall system structure.
  + The Client and Server side are purposefully kept separate, which enables each piece to develop independently from each other
  + Offers portability of the system across these systems, and the system is layered, each one constrained to that specific component and its corresponding behavior.
* For Storage management specifically, it should be used to store data and information that is not needed to be accessed quickly or modified immediately. Storage is most beneficial when being used for:
  + User account information
  + The teams registered
  + The members within each team
  + The Team play history.
* Allocating storage management using linked lists can help keep the data organized and help prevent external fragmentation when data is updated, changed, or removed. This will allow for the maximum benefit of storage space and management.
* Draw It or Lose It will use its memory management to temporarily store data. As the data is only stored temporarily, it is also accessed quickly. The rapid deployment during a game session when images are being flooded to the screen and score keeping only needs to be temporary until a team wins or loses, in which it is then stored in storage permanently.
* Security protocols is vital to protect the user’s information, the database that stores this information, as well as the internal structure of the game.
  + Multiple Authentication when a user logs in will help ensure that the proper user is accessing their profile and game.
  + Using a one-time code if the device is new or unknown, is another layer of protection for the user.
  + Requiring complex passwords will help prevent malicious access to these accounts and the database.
  + Allowing Microservices to help create an account and authenticate and individual is separate and independent from the application itself which makes it easier to communicate over IP networks.