

Draw It or Lose It

# **CS 230 Project Software Design Template**

Thomas Corrigan

Version 1.1

## 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 | 9/25/2021 | Thomas Corrigan | Creation for Module 3 |
| 1.1 | 10/15/2021 | Thomas Corrigan | Modification for Module 5 |

**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 software design problem is that Creative Technology Solutions (CTS) has taken on a new client, The Gaming Room, who needs a web-based game that can run on many platforms, but they currently only have an Android application. The online market is much bigger than the Android-only space that The Gaming Room has been serving currently so CTS is prepared to solve this problem. The solution is to create a web-based game which can run on any system, regardless of operating system of browser. Essentially the Android game needs to be created in Java so that it can run on a website from any device. It is unclear if the Android game was Java, C++, or another language.

## [Design Constraints](#_2et92p0)

The design constraints for this project are that this game must resemble the successful game that The Gaming Room created for Android devices, which is called “Draw It or Lose It”. Teams need to guess what is being drawn from a library of images and as it is drawn the players need to take guesses until someone gets it right. There will be four one-minute rounds in the game. Drawings are fully rendered at the 30-second point. After the first minute, if the teams who’s turn it is did not correctly guess the drawing then the other teams take turns guessing within 15 seconds. We are left with the following software requirements (Project One Guidelines and Rubric, 2021):

* 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.

## [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 relationship between the Entity class and the Game, Team, and Player classes is an “is-a” relationship. This means a Game is an Entity, a Team is an Entity, and a Player is an Entity. All three are inherited from Entity. They all have two common attributes, id and name. They also share public methods and one private method. The relationship between GameService, Game, Team, and Player classes is a “has-a” relationship. This is because the GameService class has Game and a Game class has a Team and a Team class has a Player. The 0…\* means a Game class could have zero to many Teams and a Team class could have zero to many Players or a Game Service class could have zero to many Games. The SingletonTester class has one method testSingleton() and it is used by ProgramDriver class which is where the main() method resides.

**"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 not the best option for server side application because of the lack of support and services running it. There are terminal commands which could be useful but it would not be an out-of-the box solution. | Linux is proven as a preferred server solution by many businesses. There are many commands available to do anything the game service would need to do and it is open source and considered very secure. | Windows has more web hosting services available and it is very easy to use compared to other operating systems and there is a large amount of software already available for Windows. | Using mobile devices as a server side solution has more of a decentralized option. Each device could connect to each other device, but security would not be as high as a dedicated central Windows or Linux server and support would be difficult. |
| **Client Side** | It is more difficult to find experienced Mac programmers compared to Windows or Linux but there is a large portion of the population who uses Mac devices so it is important that the software runs on these devices regardless of browser or version. It would probably cost the most of all four. | Linux requires the least cost but the most time and expertise compared to the other options. | Windows requires a medium cost and low expertise requires to support multiple clients. There are more people skilled in Windows to choose from. | The cost of mobile device software development is similar to Mac and so is the time and expertise required but the amount of people who use mobile devices make this a very important category. |
| **Development Tools** | Macs are capable of running Java, Python, and PHP like many other machines. There are some IDEs available but it is not the most popular development environment. | Linux is a very development friendly operating system with several IDEs available and many versions of IDEs ported to Linux. Linux is capable of running Java, Python, or PHP as well as many other languages. | Windows has many IDEs available and has been proven as a very useful operating system for development. Visual Studio and Eclipse are two of the most popular that I have used. | Mobile deivces are capable of running Java, Python, and PHP but the IDEs for mobile devices are usually run on Windows or Linux. |

## 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 would recommend the operating platform in the Windows environment. There is a wide range of support for Windows servers and most computer users use Windows so it will be familiar to anyone using it. There is also a lot of support available for Microsoft customers when compared to other operating systems such as Linux. Because Linux is free, each distribution has limited resources available for customer support.
2. **Operating Systems Architectures**: Windows has many IDEs available for developing software an many languages but the architecture that is chosen should be one that will run on every device which is why I would recommend Java. Java is compatible with millions of devices around the world, and it has such high compatibility because all of the code runs in a virtual machine which allows any device to run the same virtual machine regardless of the native architecture of the device.
3. **Storage Management**: For storage management, the game data has a couple of different options. The data could be stored server-side, client-side, or neither if a service such as Amazon cloud data services would be used. The benefits of keeping data in a server is that it is faster an more reliable than a cloud data service. Cloud data services are scalable but can also be limited in terms of bandwidth and accessibility depending on the provider so in this case I am recommending server-side storage. This storage should be in a multi-drive RAID configuration to protect against possible outages if the storage device fails.
4. **Memory Management**: The windows platform has built-in memory management techniques such as swapping, paging, and virtual memory. Swapping memory is when the process is temporarily swapped out of the main memory and into the backing store. It is then brought back into memory when it is needed. Paging suffers from the problem that different size memory chunks are difficult to fit in the backing store so the technique of paging is used to fit memory into pieces with a fixed size so that it can be handled more efficiently. The data is stored in a page table and used when needed by the CPU. Windows servers also use virtual memory which is useful because the process can be larger than the amount of RAM installed in the computer.
5. **Distributed Systems and Networks**: Distributed systems such as the connections between the players and the server will be created based on the client/server method. The players will each have a client application which will authenticate with the server and the server will have a server application which will store player game data in its storage space and retrieve this data to send back to the connected client application. Connectivity must be done through HTTP, so an authentication system will need to be developed for secure connections to the server. Another important consideration is the possibility of outages which can be mitigated through other means such as implementing a RAID storage array. The game will be completely dependent on the server in the client/server design pattern.
6. **Security**: Windows comes with a built-in antivirus which is to keep the server safe. There is also a wide range of available security software available for Windows such as penetration testing software. This software can be used to look for security flaws in web-based systems. This is an essential step before any release of software onto the internet. The client will need to authenticate to the server through HTTP so it would be useful to consider HTTPS for secure communication. Without it, anyone could track data being sent to and from the server which could lead to unauthorized access of user accounts.

## [Evaluation – Continued (Module 5)](#_2o15spng8stw)

The game, “Draw It or Lose It” can be expanded into a web application and it can gain the ability to function on IOS devices as well by following several recommendations. The first recommendations to discuss pertain to the server application. This application has a variety of available choices for which operating system to use for the server. For the web application to function across multiple recommendation systems the server must be designed to allow it. One of the most common choices for operating systems is Linux.’ Linux is “considerably more secure than Windows” (Windows, Mac, or Linux?, 2019) although no operating system is completely secure, Linux is first in the race. This can be attributed to the fact that Linux it is open source. If all the source code is publicly available, anyone can help make it more secure in future versions. Linux is also free of cost and there is no subscription to have to keep paid every year, which is one of the largest benefits of going with this option. Linux is also a good choice because there is plenty of server software to assist with hosting web pages so it would be easy to develop a web application and host it on that. Linux can also run a variety of programming languages and host them for other computers securely. The biggest downside which makes Linux a less popular choice is that it is more difficult to use and has less support since there is no company behind it.

Windows is the next obvious possible choice for a server. Windows is the most common operating system available. There is also a central company operating which provides support to customers if they need it This can be useful if there is ever a problem with the operating system and assistance is needed from the manufacturer. Windows is also easy to use and compatible with a very large collection of software and it is incredibly easy to use. If money was not a constraint, then it might be easier to set up but Windows is not without downsides. Windows is vulnerable to security threats including viruses and malware which could compromise your files. Because of this, any recommendation for Windows should also include the purchase of an enterprise antivirus program. Windows is so notoriously unsafe that it should not be run without one. The high cost of the Windows operating system as well as the yearly cost of keeping it safe makes Windows a less attractive choice than Linux but I believe it is the best choice because it is easy to use, support is available, and it requires very little expertise to get functioning properly and securely.

Mac is worth mentioning because it would be possible to use on a server although it is not recommended because they have no dedicated version of Mac for servers. It is important to use an operating system distribution designed for servers so that it is easy to configure and protected from security threats. Mac is slightly more secure than Windows but the ease in setting this one up makes it less than optimal. In summary, Linux would be the most secure server and Windows would be the easiest to configure but Linux is free and Windows is rather expensive comparatively.

The next type of recommendation to be discussed is the client-side considerations. For the client application, the software will have to run on IOS, Android, and desktop computers running Windows, Linux, or Mac. Several options are available for client software such as Java, running most of the program on the client machine while grabbing data from the server or PHP, which could run the main program on the server while simply updating the user’s web page as a front end.

Java is an easy choice because it runs in a virtual container so it can run on any of a wide range of devices. Any device with Java installed would be able to run a java application hosted by the server. The amount of time it would take to implement Java would be very little depending on which programming language the original “Draw It Or Lose It” was created with. If it was made for Android devices using Java, then it would be very easy to change it to a program that could run on any other device. Java would also be very cheap to implement because the amount of expertise required is not extensive and much of the original code could be reused.

PHP is also a good choice, but it does have it’s own drawbacks. It would be more difficult to implement a PHP frontend because the backend program will also have to be developed from the ground up. It would take more expertise to set up a program running C or any other programming language on the server and then connecting to it using PHP on the client machine. Because of this, it would presumably be more expensive. It would also take more time to redevelop the application instead of just copying most of it from Android to a java container on the client machine. This option also requires more bandwidth than running a Java program on the client machine because most of the computations would have to happen on the server with the client just waiting until it is their turn for their share of server resources. Because of the difficulty and added time to completion and possible bandwidth issues, PHP and server-side programming is not recommended.

The programming tools required for the client and server applications depend on which choice is used. If Linux is the server operating system, then there are a variety of applications that could be used depending on which language is used to create the application. Eclipse is a popular IDE which can edit Java code and it is available for Windows as well. If Windows is chosen, there are a wide variety of applications including Eclipse and Visual Studio. For the client-side programming, Eclipse could be used again for the PHP or Java code that will need to be developed or modified from the existing application. If PHP and a front-end/back-end application would need to be set up then it might be beneficial to have two teams of software developers to focus on the two different types of applications. Another popular choice would be NetBeans. Both of these IDEs are free so it would be up to the developer’s personal preference which would be best for them to use.

## Sources

Project One Guidelines and Rubric. 2021. Southern New Hampshire University. Retrieved from <https://learn.snhu.edu/d2l/le/content/850378/viewContent/14719880/View>

Windows, Mac, or Linux? We compare the pros and cons of these computing platforms. April 8, 2019. David Gewirtz. ZDNet. Retrieved from <https://www.zdnet.com/article/windows-mac-or-linux-we-compare-the-pros-and-cons-of-these-computing-platforms/>