

Draw It or Lose

# **CS 230 Project Software Design Template**

Version 1.0

Mark Maddon

Southern New Hampshire University

## 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 | 05-22-2021 | Mark Maddon | In this new revitalized game, there have been additions of attributes and methods with constructors to give the ability to have one or more teams involved.  Also, each team will have unique multiple players assigned to it and will be able to choose a team name.  Only one instance of the game can exist in memory at any given time by creating unique identifiers for each instance of a game, team, or player. |

## [Executive Summary](#_sbfa50wo7nsh)

The current software did not have the ability to have one or more teams involved. The users were requiring that each team will have multiple players assigned to it. They also wanted to implement game and team names in the program, and they must be distinctive to allow users to check whether a name is in usage when deciding on a team name. There must be the stipulation that only one occurrence of the game can exist in memory at any given time. This can be accomplished by creating unique accessories for each setting of a game, team, or player.

## [Design Constraints](#_2et92p0)

* The problem the users/staff members at The Gaming Room have is they do not know the specifications on setting the environment for a successful program. In confronting the development of the web-based version of the gaming app, integrated development environments such as Eclipse or IntelliJ can help in streamlining the development.
* While addressing their software requirements, a confirmation that the business itself supports and promotes the additions and changes to the program will be necessary. Secondly, having a specific plan and direct comments on the program to explain changes and parts of the structure created will be essential. And lastly, using proper project techniques the hardware requirements will come later because of the software application decisions. In utilizing the OS of Windows
* In addressing the needs of the clientele, the users/company will need to have specific comments made to have a sensibility because certain parts of the program needed additional attributes and methods in combination with new constructors. Therefore, in having these explanations they should be able to understand the new environment. By having these understandings, in the future more upgrades or changes within the program should be less taxing (Lucidchart, 2017).

## [System Architecture View](#_ilbxbyevv6b6)

The topology used would be Network. It would control how the computers in a network are binded to each other if they are utilizing an OS from either a Windows to a Linux or a Mac. In the different contexts of a game, the topology regulates how the OS participating in the game will be arranged to provide all players/users with an up-to-date version of the game (Network Topologies and Sample Games - Multiplayer Game Programming: Architecting Networked Games, 2016).

## [Domain Model](#_8h2ehzxfam4o)

In the UML class diagram, each of the classes have both attributes and methods except for the ProgramDriver and the SingletonTester. They both can exist without the other classes including the Parent or Superclass. The ProgramDriver is dependent on having the SingletonTester to have one line of connection. The entity, which is the Parent, is connected by inheritance to all the three child subclasses: Game, Team, and Player. The child subclasses are connected a multiplicity of zero to many and the GameService is reflective off Game in the attributes and methods it needs to function properly (Lucidchart, 2017).

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Macs are easier to install and update and allows different applications to be introduced and managed with greater simplicity in hosting a web-based software application. However most Macs don’t come with the capability of being upgraded. Those that do require complex and long processes to accomplish.  This can be handled by one development team member or multiple development team members. If any completed original product is created, there may be some copyright finances involved to be able to own whatever product entirely. | Linux is a choice for those considering utilizing Python, Ruby, or other programming languages with Open Source being the software type of use.  Pros: rarely dealt with cyber criminals; Fewer demands on Hardware  Cons: Not all versions come with long-term support and lots of professional programs are not compatible with Linux. This can be handled by one development team member or multiple development team members. If any completed original product is created, there may be some copyright finances involved to be able to own whatever product entirely. | Windows programs are available in the form of graphic user interfaces (GUI). The user has most of the control over all the basic hardware resources, receives major commentary from the OS, and can also install software automatically.  Pros: Good for users that are just starting out; upgrades on software are easily available  Cons: open to spam and other forms of malware; This can be handled by one development team member or multiple development team members. If any completed original product is created, there may be some copyright finances involved to be able to own whatever product entirely. | Mobile devices can be utilized anywhere if there is a WIFI connection available. To setup a new user or game player, provide a complete URL, provide a username and password and the player is ready to begin playing the game. Any software on the device will always be up to date. However, if the mobile device loses its frequency the player could lose his or her place in the game. Mobile devices are more fragile and require protection to not be damaged. This can be handled by one development team member or multiple development team members. If any completed original product is created, there may be some copyright finances involved to be able to own whatever product entirely. |
| **Client Side** | Macs are more expensive than other operating systems but are still good OS for beginners. They can be known to be able to run any application, which can apply to Windows and Linux and help for convenience. | Linux has more free applications than Macs or other OS and are also more secure than Windows because it takes on user permissons. Linux OS need upgrades and users can choose the time on when it needs to happen. | Windows OS are universal especially when it comes to distribution from within other computers. They are very beginner oriented. The cost for Mac can be also under $200, which is a lot cheaper than other OS. | Mobile devices’ cost can differentiate, depending on what model, and dictate whether a beginner with the game can access it as easily. If the signal is lost, the game can discontinue, which less the attraction of the game at hand which can be frustrating for beginners to want to continue. |
| **Development Tools** | Eclipse and Android Studio can be used for creating software on Macs. Also, IntelliJ IDEA can be utilized to build for Mac OS. Java and Python languages can be utilized to create this type of gaming program. | Python, Eric, and IntelliJ IDEA can be utilized to created game software. Python and C++ are languages that can create game software and programs | Eclipse and IntelliJ IDEA can be utilized to build for Mac OS. Java and Python languages can be utilized to create this type of program and gaming software. | Python, Java, and C++ are programming languages on mobile devices for gaming software. Android is used by mobile devices to run, and test completed programs. |

**(Linux vs. Windows: A Comparison of the Best Web Server Solutions, 2021).**

## 

## Recommendations

1. **Operating Platform**: Windows is the appropriate operating platform. By utilizing IDE’s that are compatible with Windows, Draw It or Lose can be distributed to other computing environments.
2. **Operating Systems Architectures**: 64-Bit architectures are used with the chosen operating platform. 64-Bit systems can utilize a lot more memory than 32-Bit architectures. They are also known to have greater speed. Games can need more space in memory, which 64-Bit is capable of. 64-Bit has extra security and “kernel patch protection,” which can help to continue a game to run safely (32-Bit vs 64-Bit, n.d.).
3. **Storage Management**: The Windows Storage Management provider controls a large amount of storage arrangements, from “single-disk desktops to external storage arrays” (Windows Storage Management Provider, 2018). Other Windows OS can support Windows-based storage management for their products by executing a Storage Management Provider (SMP) (Windows Storage Management Provider, 2018).
4. **Memory Management**: Each process from within Windows must have enough R.A.M. to configure the current program, and it can neither run into the memory space of another OS that can disrupt the current game. “The different types of memory in the system must be used properly so that each process can run most effectively” (Coustan, 2020).
5. **Distributed Systems and Networks**: Windows is fully compatible with another version of Window OS. The work with Windows is more reliable than Linux. Linux sacrifice’s reliability to be more of an open source. Because of that compatibility, distribution between other Window OS connected to games can remain connected with fewer problems (Linux vs. Windows: A Comparison of the Best Web Server Solutions, 2021).
6. **Security**: In having a secure networking with Windows, it looks towards the network that is available to perform its appointed task within the game. Its need is to protect the program/game both internally and externally (Warren, 2005).

Windows does put forth the special requirements needed “including user authentication, user device protection and point solutions.” However, the more connectivity the program has with other OS, there can be new exposure to viruses or other vulnerabilities. Therefore, alternate users can be a part of the attack from the network than just the network itself (Warren, 2005).

If other users connect over the network with other users, the connection must be delivered safely, “avoiding threats such as theft of intellectual property or private data.” Also, the fundamental foundation of the game must be protected against service maladies and disturbances, which can be ran into via network from outside sources (Warren, 2005).

References:

*32-bit vs 64-bit*. (n.d.). Diffen. Retrieved May 23, 2021, from <https://www.diffen.com/difference/32-bit_vs_64-bit>

Coustan, C. F. D. (2020, October 22). *How Operating Systems Work*. HowStuffWorks. <https://computer.howstuffworks.com/operating-system7.htm>

Emms, S. (2020, January 3). *21 Best Free Linux Integrated Development Environments (IDEs)*. LinuxLinks. <https://www.linuxlinks.com/ide/>

*Fed Up With Windows 10? Here are 12 Reasons Linux Might Be Better*. (n.d.). Lifewire. Retrieved May 23, 2021, from <https://www.lifewire.com/windows-vs-linux-mint-2200609>

Fox, L. (2020, May 15). *Advantages And Disadvantages - Web Apps*. Objective Computing Ltd. <https://objectiveit.com/blog/the-advantages-and-disadvantages-of-web-apps/>

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

*Linux vs. Windows: a comparison of the best web server solutions*. (2021, January 14). IONOS Digitalguide. <https://www.ionos.com/digitalguide/server/know-how/linux-vs-windows-the-big-server-check/>

Lucidchart. (2017, July 21). *UML Class Diagram Tutorial*. YouTube. <https://www.youtube.com/watch?v=UI6lqHOVHic>

*Open Source Software Directory*. (n.d.). Open Source Software Directory. Retrieved May 23, 2021, from <https://opensourcesoftwaredirectory.com/App-development/IDEs/MAC_OS_X>

*Network Topologies and Sample Games - Multiplayer Game Programming: Architecting Networked Games (2016)*. Computer Science, Programming, Web, Software. <https://apprize.best/programming/game_3/7.html#:%7E:text=A%20network%20topology%20determines%20how,version%20of%20the%20game%20state>.

SearchCIO.com Staff. (2007, September 1). *Five ways to streamline your application development process*. SearchCIO. <https://searchcio.techtarget.com/podcast/Five-ways-to-streamline-your-application-development-process>

Team, I. (2019, October 23). *13 Best Mobile IDEs for Android [Fall 2019 Update]*. Intellectsoft Blog. <https://www.intellectsoft.net/blog/5-of-the-best-mobile-ides-for-android/>

Warren, P. N. C. (2005b, September 29). *Ten steps to secure networking*. Computerworld. <https://www.computerworld.com/article/2559866/ten-steps-to-secure-networking.html>

*Windows Storage Management Provider*. (2018, May 31). Microsoft Docs. <https://docs.microsoft.com/en-us/previous-versions/windows/desktop/stormgmt/windows-storage-management-api-portal>