



Draw-It-Or-Lose-It
CS 230 Project Software Design Document
Version 1.0

Table of Contents

CS 230 Project Software Design Template	1
Table of Contents	2
Document Revision History	2
Executive Summary	4
Requirements	4
Design Constraints	4
System Architecture View	5
Domain Model	5
Evaluation	6
Recommendations	8

Document Revision History

Version	Date	Author	Comments
1.0	03/20/2024	Andrew Emilio Di Stefano	<p>This software design document includes an executive summary, a list of requirements, a list of design constraints, a domain model consisting of a UML diagram and a description of what is depicted within it. The software design document also includes an evaluation of each operating system in terms of the server-side and client-side advantages and disadvantages, and development tools commonly used for development within each of those operating systems. Finally, the document includes a series of recommendations concerning the operating platform, operating system architectures, storage management, memory management, distributed systems and networks, and security features which should be implemented into the project.</p> <p>The code for this version of the application includes a singleton pattern for the adaptation of an ordinary class, iterator patterns to complete the “AddTeam()” and “AddPlayer()” methods, and the creation of the “Entity” base class. The code for this version also refactors the “Game” class to inherit from the new “Entity” class, the completion of code for the “Player” and “Team” classes, and the use of an iterator pattern in the “AddTeam()” and “AddPlayer()” methods.</p>

Executive Summary

The Gaming Room would like to develop a web-based game that serves multiple platforms based on their game “Draw it or Lose It”, which is currently only available on the Android operating system. Teams which participate in this game are meant to guess what is being drawn on their screens. The application will draw from a large library of stock drawings which will be shown as clues. The game consists of four one-minute rounds, and it will take 30 seconds for the image to be fully rendered. If no correct guesses are made when the one-minute time limit is exceeded, remaining teams will have a 15-second time limit to offer one guess per team. The Gaming Room needs help streamlining the development of the environment for this game.

The solution which we propose to address these design problems is a gaming environment coded in Java to serve clients on desktop computers which uses a series of appropriate system designs and design patterns to meet the requirements of The Gaming Room. These design patterns will include the singleton pattern, which is “a creational design pattern that ensures only a single instance of a class exists throughout an application” (Gheru, 2023), to make sure that only one instance of a game can exist in memory at any given time. We will also use iterator patterns to traverse collections of data (Refactoring Guru, 2024) to make sure that each game and team name selected by users is unique in an environment involving multiple teams and players.

Requirements

1. Each game must have the ability to have one or more teams involved.
2. Each team must have the ability to have multiple players assigned to it.
3. Game and team names must be unique and allow users to check whether a name is already in use when choosing a team name.
4. Only one instance of the game can exist in memory at any given time.

Design Constraints

1. The game must be developed as a web-based application. Since each game must have the ability to involve one or more teams, these teams and players will need to be connected via the internet. This means that the application must be implemented in a web-based distributed environment. The application will require some form of hosting whether through a hosting service, through a cloud service, or on servers maintained by The Gaming room themselves. Though the tools and languages used by the developers of this project will depend on the web development stack chosen for this product, the languages used in the creation of this web-based distributed environment may include Javascript, Python, Ruby and/or Java, and tools like Bootstrap, Docker, Chrome Dev tools, Git, and/or others.
2. The game must be able to serve clients on multiple different platforms. Since each team must have the ability to have multiple players assigned to it, and these users may be using Windows, Mac or Linux on their desktops, the game must be able to serve clients who use different operating systems. This means that the skills of multiple different developers with a wide range of focuses, disciplines and skills will need to be leveraged in order to serve clients across different client-side operating systems.
3. The application must include iterator design patterns. Since game and team names must be unique, the application must use a method for iterating through collections of saved game and team name data in order to determine whether or not a selected game or team name is already

in use by another game or team. This means that the developers of this application must understand how to create and implement iterator design patterns within the code which makes up this application.

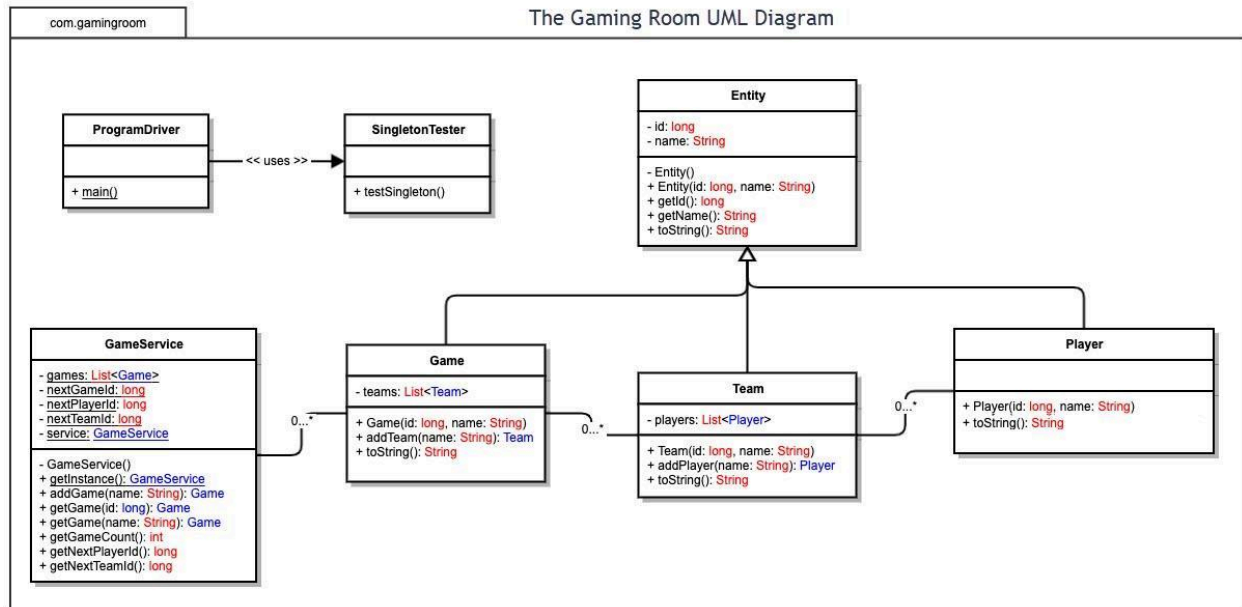
4. Each instance of a game, team or player must have its own unique identifier, and the application must use a singleton design pattern. Since only one instance of the game can exist at any given time, the application must implement a method which is capable of “ensuring that only one instance of an object exists” (Hamdouchi, 2023) in memory. The implementation of a singleton pattern is the best way to achieve this goal within the scope of this project, so the developers will have to understand how to create and implement a singleton design pattern into the application’s code.

Domain Model

The Gaming Room UML diagram depicts the interaction of different classes which will be used in this application, the types of relations which these classes will share, and the attributes and functions of each class and therefore of the objects belonging to each class. The ‘Game’, ‘Team’, and ‘Player’ subclasses (or child classes) share an inheritance relationship with our ‘Entity’ superclass (or parent class). The ‘Game’, ‘Team’, and ‘Player’ classes therefore inherit all of the attributes and methods (Lucid Software, 2027) from the ‘Entity’ class. The object-oriented programming principle of inheritance provides an efficient way for the developer to change the attributes or methods of all child classes by simply making those changes to the parent class from which they inherit those attributes or methods. In other words, if we add or remove attributes and/or methods to or from the ‘Entity’ class, these methods and/or attributes will automatically be added or removed from its child classes. This means that it is not necessary to add or remove any of the inherited attributes or methods from each of those child classes individually, making object oriented programming a highly efficient programming method.

Another relationship shown in the diagram is the association relationship which the ‘ProgramDriver’ class has with the ‘SingletonTester’ class. The association is labeled “uses” since the ‘ProgramDriver’ class uses the ‘SingletonTester’ class to test the singleton design pattern. Specifically, the public ‘main()’ function of the ‘ProgramDriver’ class uses the public ‘testSingleton()’ function of the ‘SingletonTester’ class to test the singleton design pattern to assure its functionality.

“Multiplicity allows...[us]...to set numerical restraints on our relationships” (Lucid Software, 2017). As we can see in the UML diagram, the ‘Player’ class shares a zero-to-many association relationship with the ‘Team’ class. The ‘Team’ class also shares a zero-to-many association relationship with the ‘Game’ class, and the ‘Game’ class also shares a zero-to-many association relationship with the ‘GameService’ class. This means that each team can have zero to many players, each game can have zero to many teams, and each game service can have zero to many games.



Evaluation

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

<p>Server Side</p>	<p>“Mac hosting is a virtual private server (VPS) hosting solution that provides users with dedicated resources specifically for the Mac operating system” (Idri, 2023). Using Mac OS for hosting a web-application gives the user their own isolated environment complete with infrastructure including “memory, storage, CPU, and bandwidth” (Idri, 2023) enabling the hosting of Mac applications. Mac hosting offers a robust solution for storing large amounts of data securely, and enables access to “Apple-specific features like iCloud or FaceTime” (Idri, 2023).</p> <p>One drawback to the server-side use of Mac hosting includes the relatively higher cost of compatible hardware as opposed to that of other operating systems (Sunbul, 2023). Other drawbacks to the</p>	<p>“Linux is an open-source, Unix-like operating system kernel” (Sunbul, 2023). Known for its strong security, dependable stability and high level of versatility, Linux was the operating system used for “96.3% of the top one million web servers” in 2023 (Sunbul, 2023). Linux was designed to support various users and tasks simultaneously, which is an incredibly important feature for server environments (Sunbul, 2023). Linux’s software package management is also incredibly intuitive and easy to use (Sunbul, 2023).</p> <p>One significant drawback to Linux, especially for this specific use-case, is the limited gaming support offered by Linux. “Although Linux gaming has improved significantly in</p>	<p>“Microsoft Windows Server OS...[is]...designed to share services with multiple users and provide extensive administrative control of data storage, applications and corporate networks” (Walat, 2024). One significant advantage to Windows over Linux is its “high performance web application framework, ASP.NET, whose most current version is also compatible with Linux” (Ionos, n. d.). Other benefits to using Windows as a server-side operating system to host web based software applications is its “Beginner-friendly , intuitive operations through a graphic user interface” and its ability to solve technical problems via a system recovery (Ionos, n. d.).</p> <p>Some cons to the use of Windows as a hosting</p>	<p>When considering different hosting options for web-based applications, mobile devices are not likely the first option to come to mind. The capability of mobile devices tends to be more limited compared to that of desktops or larger computers since mobile devices are meant to allow one to hold them in one’s hand during use, and usually to even fit inside of one’s pocket. Mobile operating systems like Android and iOS are made for the specific purpose of running on smaller devices and not necessarily for the hosting of web applications. That being said, the hosting of web applications on such operating systems is technically possible (Pinggy, 2024). One of few advantages to hosting a web based application on a mobile device is the lower cost of the hardware. Many mobile phones are, of course, usually much less expensive than a modern desktop computer.</p>
---------------------------	--	---	---	---

	<p>use of Mac OS to host web applications include limited customization and constraints on compatibility with non-apple hardware (Sumbul, 2023).</p> <p>In summary, Mac does offer a server-based deployment platform where the website can be hosted. That being said, Mac OS is probably not the best option for hosting the Draw It or Lose It web application. Apple no longer makes server operating systems (Apple, n. d.). The last Mac server operating system was Mac OS X Mountain Lion (Apple, n. d.). "Mac OS X 10.8 Mountain Lion is available for older systems that are not compatible with the latest [system] version" (Apple, n. d.). The licensing fee can cost \$390-\$499 depending on the reseller.</p>	<p>recent years, it still lags behind Windows regarding game availability and compatibility" (Sunbul, 2023). Other drawbacks to the use of Linux for the hosting of web-based software applications is a steeper learning curve for unfamiliar users, and the fact that "Linux may face challenges with specific hardware components, particularly those lacking official Linux drivers" (Sunbul, 2023).</p> <p>In summary, Windows does offer a server-based deployment platform where the website can be hosted. The licensing fee for Windows Server 2022 (the latest version of the Windows Server operating system) is likely to cost around \$501 for the Essential edition, \$1069 for the Standard edition, and \$6,155 for the Database version</p>	<p>solution for web-based applications are its high licensing costs, vulnerability to malware, and its limited capability of handling multi-user systems (lonos, n. d.).</p> <p>In summary, Linux does offer a server-based deployment platform where the website can be hosted. Licensing for an Ubuntu server costs \$25 for the Essential subscription type, \$150 for the Standard subscription type, and \$300 for the Advanced subscription type (Ayotte, 2022). Licensing for Linux Red Hat server costs \$349 for a one-year self-supported subscription, \$799 for a one-year standard subscription, or \$1,299 for a one-year premium subscription (Ayotte, 2022).</p>	<p>Among the vast number of weaknesses for hosting a web-based software application on a mobile device are the need to install additional software, overcoming the hurdles put into place by these very operating systems in order to make the use of mobile devices for purposes outside of what was intended by the creators, and the obvious hardware limitations of mobile devices for such an endeavor.</p> <p>In summary, there is no viable server-based deployment platform on which the website can be hosted.</p>
--	---	--	--	---

		(Microsoft, n. d.). Exact pricing will ultimately depend on the reseller's pricing for these products (Microsoft, n. d.).		
--	--	---	--	--

Client Side	<p>Though Google Chrome web browser can be downloaded onto a Mac desktop, many Mac users still use Safari as their go-to browser (Statcounter, n. d.). Safari is known for behaving strangely with certain websites and web-based applications, and assuring full functionality on Safari will likely take more time and cost than it would on Chrome, which dominates the market share by a large margin (Statcounter, n. d.). Developing native applications on Mac rather than applications which run in the browser would also likely take more time, have a higher cost, and require a certain level of expertise with Mac since the number of developers who have experience developing native Mac applications is relatively low (Kairn, 2022).</p>	<p>If Linux users are to access the application through their browsers, then the cost, time and expertise involved in supporting multiple Linux users is likely to be relatively low compared to that of Mac since Linux users are most likely using either Chrome or Firefox as their browsers. Chrome is by far the most widely used browser (Statcounter, n. d.), and since Chrome and Firefox are widely considered the best browsers for web development (GeeksForGeeks, n. d.), there are likely a large number of developers with experience developing for web applications in these browsers. Developing a native Linux application however would likely have a higher cost and require more specific expertise. Gaming support for Linux is somewhat limited</p>	<p>Windows is the most popular desktop operating system by a very large margin (Brinkmann, 2023), and most Windows users use Google Chrome browser. Since Chrome is considered one of the best browsers for web development (GeeksForGeeks, n. d.), there are likely a large number of developers with experience developing for web applications in this browser. This means that the cost and time required to serve Windows clients is likely lower than that which would be required to serve Mac users who are likely to use Safari browser. If Windows users are to access the game via a native Windows application, then the cost and amount of time involved in creating a functional and reliable Windows application would likely be</p>	<p>If users are to access the game via a mobile device, then the prospects of accessing it through a browser become much less viable. Those who play games on mobile devices are almost certainly doing so through a native application. "Mobile development is often challenging for beginners because learning the languages requires familiarity with advanced programming skills. This challenge is compounded by mobile programming languages being less versatile than some of their web development counterparts, making it difficult to rely on existing skills" Noble Desktop, n. d.). This means that developers will likely need to have to have a more significant level of expertise and experience, and that it may take more time and more monetary resources to develop a mobile application where users can access this</p>
--------------------	--	--	---	--

		<p>compared to gaming support for Windows (Sunbul, 2023). This means not only that we would be drawing from a much smaller pool of developers, but also that it would be harder to create a reliable and user-friendly native Linux application for users to access this game.</p>	<p>significantly less than if creating an application for Mac or Linux.</p>	<p>game. Luckily, The Game Room has already developed their client side Android application. In order for the game to be made available on iOS devices, the game should be developed in Swift or Objective C by a developer with experience developing applications which can run within the iOS operating system.</p>
--	--	--	---	--

Development Tools	<p>The programming language most often used to create native Mac applications is called Swift (Apple, 2024). A non-exhaustive list of IDEs currently recommended for coding in Swift by Apple on swift.org are Atom IDE, Vim, and Xcode (Apple, 2024). Some of the recommended text editors for coding in Swift are Visual Studio Code and Xcode. Visual Studio Code is free and built on open source software. Xcode is also free to download. RunSwift is also an efficient in-browser tool for writing and running Swift code (Blue Coding, 2024). RunSwift will also provide the user with a “step-by-step tutorial on the basics of Swift and the RunSwift website” (Blue Coding, n. d.). Swift is also often used to develop in-browser apps for Safari, but other common web development languages like Javascript and CSS</p>	<p>The programming languages used to write native Linux applications include C++, Java and Python. These languages are supported by a plethora of IDEs and text editors including Visual Studio, Atom IDE, Visual Studio Code, and Sublime Text. There is a free community version of visual studio, a business version which costs \$45 per month, and an enterprise version which costs \$250 per month (Microsoft, n. d.). Atom IDE and Visual Studio Code are both free and open source. Sublime Text is free to download, but “however a license must be purchased for continued use” (Sublime Text, n. d.). The Sublime Text license costs \$99 (Sublime Text, n. d.). There are a number of other tools which are used to build Linux applications such as Git (a</p>	<p>The programming languages used to write native Linux applications include C++, Java and Python, and C#. “C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework. C# is used to develop web apps, desktop apps, mobile apps, games and much more” (W3Schools, 2023). C# and the other aforementioned languages used in the creation of Windows applications are supported by many of the same tools mentioned in the cell to the left describing tools used in the creation of Linux applications including Visual Studio, Atom IDE, Visual Studio Code, Sublime Text, Git, and many others. Visual studio is the IDE recommended by Microsoft for developing Windows applications</p>	<p>The programming languages most commonly used for Android application development are Java and Kotlin (Geeks for Geeks, n. d.). While Java was originally the official language for Android development, Kotlin became the new official language for Android development as declared by Google in 2019 (Geeks for Geeks, n. d.). IDEs and text editors recommended by the Kotlin Foundation for use when developing in Kotlin include Eclipse, Visual Studio Code, and Atom IDE (Kotlin Foundation, 2023). The programming language used to create applications which run on Apple mobile devices such as iPhones and iOS Tablets is Swift. A non-exhaustive list of IDEs currently recommended for coding in Swift by Apple on swift.org are Atom IDE, Vim, and Xcode (Apple, 2024). Some of the recommended text editors for coding in Swift are Emacs and Visual Studio Code.</p>
--------------------------	---	--	---	--

	<p>may also be used to develop apps meant to run in a Safari browser. Objective C is another app which is sometimes used for developing Mac OS applications, though the majority are written in Swift (Galiya, 2023). Development team members or teams responsible for producing Mac OS applications should have professional proficiency with these languages and experience working in Mac OS environments.</p>	<p>popular version control system), Inkscape (a vector graphics editor) (Malik, 2023), and Virtual Box (an application which can be used to create Linux virtual machines where applications can be tested safely). Git is free to use and download, but additional functionality may be added for a fee. Github, a cloud-based hosting service which allows users to host git repositories online, has three plans for development teams. There is a Free version, a Team version which costs \$3.69 per month per user, and an Enterprise version which costs \$19.25 per user per month at the time of writing (Github, n. d.). Inkscape is free to use, and Virtual Box is also free to use for individuals. Virtual Box Commercial Licenses cost \$50 per user with a</p>	<p>(Microsoft, n. d.). "The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app" (Microsoft, n. d.). There is a free community version of visual studio, a business version which costs \$45 per month, and an enterprise version which costs \$250 per month (Microsoft, n. d.).</p>	<p>While single developers have successfully created mobile apps, a project could greatly benefit from having a team consisting of professionals with different experience and skill levels with different stacks of the aforementioned tools.</p>
--	--	--	--	--

		minimum of 100 users (Redress Compliance, 2024), which comes out to at least \$5000 for the minimum number of users .		
--	--	---	--	--

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:

The operating platform which we recommend to our client The Gaming Room for their game “Draw It or Lose It” is Windows Server 2022 on a virtual Windows server instance within the Microsoft Azure cloud with their Platform as a Service (PaaS) development and deployment environment (Microsoft, 2024). There are several reasons for this recommendation.

If The Gaming Room were to maintain their own physical servers, then the scalability of the “Draw It or Lose It” application would depend on The Gaming room’s ability to purchase and set up new servers. For example, if the number of players were to increase rapidly in a short period of time, then the platform in place might struggle to serve this larger number of clients. This could have a negative effect on the application’s performance and therefore on the ability for The Gaming Room to retain these new clients. In such a situation, the Gaming Room would have to acquire and set up new servers in order to scale up as quickly as possible. This would of course take time and monetary resources, and if the number of users were to go back down then this new infrastructure would no longer be needed. Cloud platforms like Azure facilitate the use of a powerful cloud function called “load balancing,” which allows web applications to spread workloads across different virtual servers as needed. This way, The Gaming Room can easily scale up or down in order to control expenses while maintaining constant and reliable functionality for their users.

While it is possible to run Windows servers on other cloud platforms such as that of Amazon Web Services, we recommend Azure specifically because like Windows, Azure is a product of Microsoft. Windows is a proprietary operating system, and it is of course in the best interest of Microsoft to offer more advantages to those who use their cloud service as well. For example, Microsoft Azure “offers native support for Windows Server workloads” (Microsoft, 2024). Azure also offers “up to 85 percent compared to standard pay-as-you-go rates...[on]...existing Windows Server...licenses” (Microsoft, 2024).

For these reasons, we recommend the use of Windows Server 2022 on a virtual Windows server instance within the Microsoft Azure cloud as the ideal operating platform for our client The Gaming Room’s web application “Draw It or Lose It.”

2. **Operating Systems Architectures:**

“The architecture of Microsoft Azure comprises data centers housing physical servers, virtualized hardware, and a complex network infrastructure, facilitating the deployment and operation of cloud-based applications and services” (Simplilearn, 2024). The scale, network redundancy, and sheer power of this type of cloud infrastructure and architecture assure that the “Draw It or Lose It” application will be constantly available to clients, and the scalability and efficiency achieved through the use of load-balancing and other virtual server optimization methods means that “Draw It or Lose It” can grow (or shrink) as required as users start and finish their gaming sessions. By leveraging the power of Microsoft Azure, The Gaming Room will be ready to handle the unpredictable nature of the gaming market. The level of support offered by Microsoft to Microsoft Server on Azure clients will also be a powerful resource helping to assure that The Gaming Room’s “Draw It or Lose It” application is set up in a way which optimizes Azure and Windows Server capabilities while making the system as capable and efficient as possible.

3. **Storage Management:**

Azure offers efficient storage management and provides built-in measures for accessing stored information. Azure cloud storage management allows clients to “create, delete, view, edit, and manage resources for Azure Storage, Azure Data Lake Storage, and Azure managed disks” (Microsoft, 2024) through their intuitive user interface which allows users to “view data effortlessly and quickly” (Microsoft, 2024). Azure Storage Explorer is highly customizable and allows the user to add extensions as needed, including the Azure Data Factory extension used to move data from other storage systems to Azure Storage, and the Azure App configuration extension which allows admins to manage application settings and feature flags from one convenient interface (Microsoft, 2024). Azure Storage Explorer also allows clients to work while disconnected from the cloud or even while disconnected from the internet itself through the use of local emulators such as Azurite, allowing users to “boost...productivity and efficiency while reducing costs” (Microsoft, 2024).

4. **Memory Management:**

Azure allows memory management to be customized based on user requirements and needs. “Azure Stack is built as a hyper-converged cluster of compute and storage. The convergence allows for the sharing of the hardware, referred to as a scale unit...a scale unit provides the availability and scalability of resources...[consisting]...of a set of Azure Stack servers, referred to as hosts or nodes” (Microsoft, 2024). Scale units can be set up within Azure so that the amount of memory which is needed at any given time can be added in order to meet the needs of an increasing number of users, or subtracted in order to keep costs down during times when less users are interacting with the system. This type of storage scalability can be achieved by configuring virtual machines “in an availability set that spreads them across multiple fault domains” (Microsoft, 2024). “If there is a host failure, VMs from the failed fault domain will be restarted in other hosts, but if possible, kept in separate fault domains from the other VMs in the same availability set. When the host comes back online, VMs will be rebalanced to maintain high availability” (Microsoft, 2024). The use of a network of virtual servers drawing resources from a series of physical servers operated and maintained by Microsoft will allow The Gaming Room’s application “Draw It or Lose It” to set up an efficient and scalable memory system which provides redundant measures for memory maintenance in the event of host failure, and which is

capable of scaling memory up to account for an influx of users, and of scaling memory down when less users are connected to the system.

5. Distributed Systems and Networks:

By using distributed software and networks run through the Azure cloud, Draw It or Lose It will be able to communicate between various platforms, constantly accessible to clients using different operating systems and platforms to access the application. The fact that the services and functionalities used by Azure virtual machines can come from any of Microsoft's physical servers which make up their vast cloud network physical machines around the world means that if one building or area were to experience a network disturbance or power outage, other machines which make up the network controlled and maintained by Microsoft would be able to pick up the slack and keep clients' server instances up and running. While this solves potential issues related to connectivity and outages which could take place on the server side, clients who would like to connect to the "Draw It or Lose It" application will of course be responsible for the power and internet connections of their own devices on the client side. As long as these clients are able to keep their machines powered and connected to the internet, the application should be able to communicate with their operating platforms. In most of the areas in which potential customers will be targeted, maintaining power and a reliable internet connection on the client side should not be a problem.

6. Security:

Windows Server 2022 offers "secured-core capabilities to help protect hardware, firmware, and Windows Server OS capabilities against advanced security threats...such as Windows Defender System Guard and Virtualization-based Security to minimize risk from firmware vulnerabilities and advanced malware" (Microsoft, 2024). Secured-core is meant to simplify security for ease of use by admins while providing advanced protection through both reactive and preventative defense measures (Microsoft, 2024). "Secured-core Servers come with the assurance that manufacturing partners have built hardware and firmware that satisfy the requirements of the operating system...security features" (Microsoft, 2024). The latest Windows Server operating system on Azure also provides "secured connectivity...[with]...secure encrypted HTTPS connections...[and]...industry standard SMB AES 256 encryption" (Microsoft, 2024). User information on and between platforms will be protected through the robust security measures implemented by Microsoft through Microsoft Server OS and Azure. While leveraging the robust security offered by Microsoft through Windows Server Azure, The Gaming Room will certainly not be on their own in terms of security.

Sources:

- Apple. (2024). *Swift*. <https://www.swift.org/>
- Ayotte, L. (2022, October 26). *How Much Does a Linux Server Cost?* SeverMania. <https://www.servermania.com/kb/articles/how-much-does-a-linux-server-cost>
- Blue Coding. (2024). *The 5 Best Tools For Swift Developers*. <https://www.bluecoding.com/post/the-5-best-tools-for-swift-developers>
- Brinkmann, M. (2023, February 8). *Windows 10 continues to be preferred by PC gamers*. Ghacks. <https://www.ghacks.net/2023/02/08/windows-10-continues-to-be-preferred-by-pc-gamers/>
- Apple. (n. d.). *Mac OS X Mountain Lion Installer*. <https://support.apple.com/en-us/106387>
- Dilger, D. E. (2011, June 18). *Apple Insider. Apple reveals Mac OS X Lion, Server prices for business and education*. Apple Insider. https://appleinsider.com/articles/11/06/18/apple_reveals_mac_os_x_lion_server_prices_for_business_and_education
- Downs, C. (2024, February 13). *How much does a server cost for app hosting?* ServerMania. <https://www.servermania.com/kb/articles/server-cost-for-apps>
- Galiya, J. (2023, December 17). *Diving Deep into iOS Programming Languages: Advanced Techniques for iOS App Development*. Radix. <https://radixweb.com/blog/best-programming-languages-for-ios-app-development>
- Geeks for Geeks. (n. d.). *Top Programming Languages for Android App Development*. <https://www.geeksforgeeks.org/top-programming-languages-for-android-app-development/>
- Geeks for Geeks. (n. d.). *What are the best Browsers for Web Development?* <https://www.geeksforgeeks.org/what-are-the-best-browsers-for-web-development/>
- Ghergu, E. (2023, June 27). *The Singleton Design Pattern*. PentaBlog. <https://www.pentalog.com/blog/design-patterns/singleton-design-pattern>
- Github. (n. d.). *Get The Complete Developer Platform*. <https://github.com/pricing>
- Hamdouchi, M. (2023, June 1). *Singleton Design Pattern: Ensuring Only One Instance of an Object Exists*. Medium. <https://medium.com/@mhamdouchi/singleton-design-pattern-ensuring-only-one-instance-of-an-object-exists-f7d0f89d357a>

- Hyde, T. (2023). *Best Programming Languages for Linux Development*. Black Down.
<https://www.blackdown.org/best-programming-language-for-linux/>
- Idri, D. (2023, March 19). *What is Mac Hosting, Benefits, How to get started?* UltraHost.
<https://ultahost.com/blog/what-is-mac-hosting-benefits-how-to-get-started/>
- Ionos. (n. d.). *Linux vs. Windows Servers*.
<https://ultahost.com/blog/what-is-mac-hosting-benefits-how-to-get-started/>
- Kairn, P. (2022, March 11). *Why coding macOS apps is hard + 3 learnings from @JulienLacr0ix to crack it*. Medium.
https://medium.com/@patricia_11833/why-coding-macos-apps-is-hard-3-learnings-from-julienlacr0ix-to-crack-it-313f2dece56
- Kotlin Foundation. (2023, November 20). *IDEs for Kotlin Development*.
<https://kotlinlang.org/docs/kotlin-ide.html>
- Lucid Software. (July 21, 2017). *UML Class Diagram Tutorial* [Video]. Youtube.
<https://www.youtube.com/watch?v=UI6lqHOVHic>
- Malik, N. J. (2023, January 230). *Top 10 Linux tools and apps for developers and system admins*. Codedamn.
<https://codedamn.com/news/linux/top-10-linux-tools-and-apps-for-developers-and-system-admins>
- Micron Technology. (2024). *Mac vs. PC: The Pros and Cons*.
<https://www.crucial.com/articles/mac-users/mac-vs-pc-the-pros-and-cons>
- Microsoft. (2024). *Azure Storage Explorer*.
<https://azure.microsoft.com/en-us/products/storage/storage-explorer>
- Microsoft. (n. d.). *Github Copilot and Visual Studio Code*. <https://visualstudio.microsoft.com/>
- Microsoft. (n. d.). *Pricing and licensing for Windows Server 2022*.
<https://www.microsoft.com/en-us/windows-server/pricing>
- Microsoft. (2024). *Virtual machine memory allocation and placement on Azure Stack*.
<https://azure.microsoft.com/en-us/blog/virtual-machine-memory-allocation-and-placement-on-azure-stack/>
- Microsoft. (2024). *Windows Server on Azure*.
<https://azure.microsoft.com/en-us/free/windows-server-on-azure>
- Microsoft. (2024). *What is PaaS?*

<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-paas>

Microsoft. (2024). *Windows Server 2022*.

<https://www.microsoft.com/en-US/evalcenter/evaluate-windows-server-2022>

Noble Desktop. (n. d.). *How Difficult is it to Learn Mobile Development?*

<https://www.nobledesktop.com/learn/mobile-development/how-difficult-is-it-to-learn-mobile-development>

Pinggy. (2024, February 1). *You can host your website on Android! Here's how.*

https://pinggy.io/blog/host_website_on_android/

Redress Compliance. (2024, February 29). *Oracle Virtualbox License - Navigate Free and Paid License.*

<https://redresscompliance.com/oracle-virtualbox-license-navigating-free-and-paid-license/>

Refactoring Guru. (2024). *Iterator*. <https://refactoring.guru/design-patterns/iterator>

Simplilearn. (2024, March 25). *Azure Architecture Explained.*

<https://www.simplilearn.com/azure-architecture-explained-article>

Statcounter. (n. d.). *Browser Market Share Worldwide.*

<https://gs.statcounter.com/browser-market-share>

Sunbul. (2023, December 20). *Linux vs Mac: 9 Crucial Factors to Consider.*

RedSwitches. <https://www.redswitches.com/blog/linux-vs-mac/>

W3Schools. (2023). *C# Tutorial*. <https://www.w3schools.com/cs/index.php>

Walat, T. (2024). *Microsoft Windows Server OS (operating system)*. TechTarget.

<https://www.techtarget.com/searchwindowsserver/definition/Microsoft-Windows-Server-OS-operating-system>