



Draw It or Lose It
CS 230 Project Software
Version 1.2

Table of Contents

CS 230 Project Software	1
Version 1.0	1
Table of Contents.....	2
Document Revision History	2
Executive Summary	3
Requirements.....	3
Design Constraints	3
System Architecture View.....	3
Domain Model	4
Evaluation	4
Recommendations: Operating Platforms and Architectures	8
Recommendations: Storage and Memory Management	7
Recommendations: Distributed Systems, Networking and Security.....	7

Document Revision History

Version	Date	Author	Comments
1.2	Jun. 22, 2024	Alexander Ahmann	Elaborated more on the networking and security aspects of Draw It or Lose It.
1.1	Jun. 7, 2024	Alexander Ahmann	Expanding writeup to further elaborate on technical details, discuss an iOS app implementation, and elaborating on the various operating platforms.
1.0	May 22, 2024	Alexander Ahmann	Initial writeup describing the problem.

Executive Summary

Draw It or Lose It is a computer game¹ that challenges its players, the end-user, to identify an entity from an image that is slowly, but steadily, being rendered within a specified time frame. An *Android* app of *Draw It or Lose It* already exists, and the problem that we are faced with is to create a web application equivalent and Apple iOS equivalent of the *Android* app.

Requirements

From the given scenario,² I have worked out the following requirements of the project:

- A client and central service where players connect to compete with one another.
- The notion of a “team”, which is just a group of players.
 - “Teams” must consist of more than one player.
 - The ability to host *at least* one team.
- The notion of a “game”, which is an instance of a team, or multiple teams, engaging in the activity embodied by the software.³
 - Only one “game” instance can exist in-memory.
- Games and Teams must have unique names and identifiers.

Design Constraints

From the given scenario,⁴ I have worked out the following design constraints:

- The software must be accessible from a standard web browser; hence we must employ a web interface.
- In accordance with the requirements, a “team” and “game” object will be invented, and the “team” and “game” objects will be stored in some kind of Array or Array-Like data structure that allows for iterators to get information about them.
 - It must be asserted that the number of “players” in a “team” object is *greater than 1*.
 - It must be asserted that the number of items in the “team” array is *greater than or equal to 1*.

System Architecture View

While there are not many constraints regarding the system architecture, it should be noted that this implementation of *Draw It or Lose It* will be in the form of a web interface. A server component will be written in Java, implying that the system that it runs on should have a processor architecture that

¹ This computer game is derivative of *Win, Lose, Draw*, which is a game show that was popular in the latter half of the twentieth century. In it, a picture depicting a thing is painted in real time, and contestants are challenged to identify it within a time frame. The major difference between the show and our computer game is that the computer will render a stock image, whereas another contestant draws the thing on a canvas.

² CS-230 (n.d.). *Project One Guidelines and Rubric*. Southern New Hampshire University.

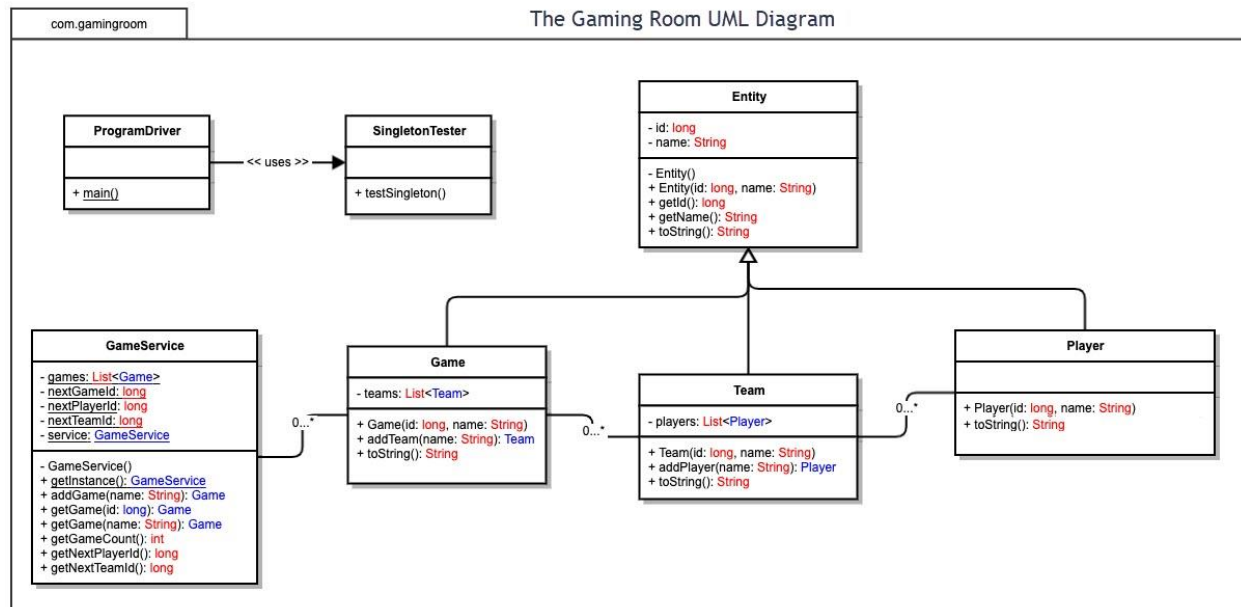
³ To avoid being redundant, I will sometimes refer to the *Draw It or Lose It* game as “the software.”

⁴ See CS-230 (n.d.).

supports the *Java Runtime Environment*. Likewise, the client-side component will be a web interface, implying that any computer system with a standard web browser that can interpret HTML and execute JavaScript⁵ should be able to run the game.

Domain Model

A *Unified Modelling Language* (UML) diagram is used to conceptualise the relationship between different entities and constructs in the Java software package embodying *Draw It or Lose It* – with the following graphic below depicting its UML diagram:



The *Player* class is used to identify an end-user who is engaging with the software. Players are added to a private list in the *Team* class, which consists of the *addPlayer()* method and a constructor. The *addPlayer()* method is parameterised with a *String* that is a unique screen name for an instance of a new *Player* object. The *Game* class is an instance of various *Teams* participating in the activity facilitated by the software service. The *Player*, *Team* and *Game* classes all inherit methods from the *Entity* class, which gives them basic shared attributes like the notion of an identifier and a unique name.

The *Draw It or Lose It* game is managed by the *GameService* class, which is an in-memory database of instances of the *Game* class in a list. This service class consists of getter and setter functions that report and change the state of a *Game* instance. Finally, the *ProgramDriver* class has the *main()* method which initialises the software, and the *SingletonTester* class is a test unit to ensure a singleton design pattern when implementing the software.

Evaluation

⁵ Note that some end-user client browsers may disable JavaScript for security or other reasons. The game developers should do their best to prevent running dynamic code on the client-side, and have the game execute entirely on the server.

The following table summarises and discusses how each component of the development requirement would function on different operating platforms (Apple macOS, Microsoft Windows, Linux distributions, and Mobile Devices), and their respective strengths and weaknesses:

Development Requirements	macOS	Linux	Windows	Mobile Devices
General Features	<ul style="list-style-type: none"> * Proprietary * Designed specifically to work with <i>Apple</i> hardware * Easy to use 	<ul style="list-style-type: none"> * Free and open source * Compatible with most mainstream processor architectures * Ease of use varies on distribution 	<ul style="list-style-type: none"> * Proprietary * Compatible with <i>x86</i>, <i>x86_64</i> and <i>AMD</i> processor architectures * May require some training to use for certain roles (s.a. systems administration) 	<ul style="list-style-type: none"> * Typically proprietary, varies by platform * Runs on <i>ARM</i> and other mobile processor architecture * Easy to use
Server Side	<ul style="list-style-type: none"> * No server-side support * Can support a <i>JRE</i> runtime for <i>Java</i> services 	<ul style="list-style-type: none"> * There are <i>Linux</i> distributions that exist to act as a web server * Can support a <i>JRE</i> runtime for <i>Java</i> services 	<ul style="list-style-type: none"> * <i>Windows Server</i> operating system is used for server-side components * Can support a <i>JRE</i> runtime for <i>Java</i> services 	<ul style="list-style-type: none"> * Not recommended to host web applications
Client Side	<ul style="list-style-type: none"> * Supports multiple web browsers: mainly <i>Safari</i> 	<ul style="list-style-type: none"> * Supports multiple web browsers 	<ul style="list-style-type: none"> * Supports multiple web browsers: mainly <i>Microsoft Edge</i> 	Web browser provided varies on operating system
Development Tools	<ul style="list-style-type: none"> * Supports the JDK * For Apple iOS app, macOS is the ideal operating platform 	<ul style="list-style-type: none"> * Supports the JDK * IDE support varies by distribution 	<ul style="list-style-type: none"> * Supports the JDK * Supports various IDEs, mainly Microsoft's <i>Visual Studio</i> or <i>Visual Studio: Code</i> 	Not recommended for software development

Discussion of Operating Platform by General Features

The *Microsoft Windows* operating system,⁶ like *Apple macOS* and some *Linux* Distributions, requires a licence to use. Despite the licencing, their server software is well made and has consistent technical

⁶ Microsoft (n.d.). *Windows Server solutions across cloud and on-premises*. Retrieved on May 22, 2024, from: <https://www.microsoft.com/en-us/windows-server>

support and software updates. Windows Server may also be easier to use when compared to Linux⁷ and the Windows operating systems are the most used out of all the desktop platforms.⁸

macOS is a proprietary operating system derived from AT&T's UNIX. Like Microsoft Windows, one needs to pay for a licence to run it. macOS may not be the ideal server operating system, as it has been discontinued.⁹ It is possible to use its legacy version, but it will lack technical support and software patches, the latter could cause issues regarding compatibility and security. Furthermore, macOS is typically restricted to Apple's proprietary hardware architecture.

Linux is a free kernel and operating system that comes in the form of distributions from a diverse network of maintainers. There is no licence system,¹⁰ but the quality of support and software updates will vary from maintainer to maintainer. Linux distributions will work on most hardware architectures, and its open-source nature will allow for system engineers and developers to "fine tune" parts of the kernel to work with the server and developer side of the software if need be.

Discussion of Operating Platform by Role in Development and Deployment

Server Side

Microsoft has a specialised operating system called *Windows Server* which, as the name implies, is designed to be installed on server platforms. Its sole purpose is to optimise resources to handle incoming connections from many client systems onto daemons running on the server, and includes features for hosting various kinds of services and enhanced security. As of the time of this writing, the current version of this operating system is *Windows Server 2022*,¹¹ which requires a licence and costs \$501 for the bare minimum "essentials"¹² and \$1069 for the "standard" version.¹³ There is a "data center" licence going at \$6,155,¹⁴ but it is unlikely that the game will need the level of resources supplied by a data center.

Most Linux distributions have their own server component. Again, because of the great diversity of Linux distributions, it is difficult to determine exactly how each distribution would handle Java services. I will opt to work with the *Debian* family of Linux Distributions, which includes *Debian Linux* itself, *Ubuntu Linux*, *Linux Mint*, and other derivatives centered on the `apt` package manager.¹⁵ Linux Distributions are

⁷ This is subjective, and system engineers should discuss whether-or-not to use Windows Server or Linux based on ease-of-use, amongst other factors.

⁸ See "statcounter" (n.d.). *Desktop Operating System Market Share Worldwide (Apr 2023 - Apr 2024)*. Retrieved on May 22, 2024 from: <https://gs.statcounter.com/os-market-share/desktop/worldwide>

⁹ Apple (n.d.). *About macOS Server 5.7.1 and later*. Retrieved on May 22, 2024 from: <https://support.apple.com/en-us/101601>

¹⁰ With some exceptions, like *Red Hat Linux*.

¹¹ See Microsoft (n.d.). *Windows Server 2022*. Retrieved on Jun. 7, 2024 from: <https://www.microsoft.com/en-us/windows-server>

¹² See Microsoft (n.d.). Pricing and licensing for Windows Server 2022. Retrieved on Jun. 7, 2024 from: <https://www.microsoft.com/en-us/windows-server/pricing>

¹³ Ibid

¹⁴ Ibid

¹⁵ See Ubuntu (n.d.). Package management. Retrieved on Jun. 7, 2024 from: <https://ubuntu.com/server/docs/package-management>

a popular operating platform for running computer servers, and with most distributions being free and open source, it will show itself to be friendly to developers. In this author's opinion, the Debian family of distributions do a particularly good job of balancing ease-of-use and convenience with customizability and security.

The Apple corporation has released their own *macOS Server* operating platform currently at version 5.7.1,¹⁶ but it has been discontinued since December 2021. Because of its lack of updates and support, I do not recommend deploying this operating system in a production environment.

While it may be possible to run a Java service on a mobile device,¹⁷ it is not recommended to do so because a mobile device operating system is most likely not to be optimised to scale, nor would it have the hardware resources to handle simultaneous connections from multiple client systems.

Client Side

All the desktop-editions of *Microsoft Windows* and *Apple macOS*, and many of¹⁸ *Linux* Distributions are ideal for running the client side of the software. They all come with a web browser that can render HTML5 and execute JavaScript, and it is fair to assume that the end-user of the client side of this software has the needed training to browse the web with their respective operating platform's web browser.

Most mobile devices have a built-in web browser that can render HTML5 and execute JavaScript, so they are an ideal device to run client-side web applications on. Furthermore, iOS and Android applications are to be developed for their respective operating platforms, so by the very nature of its operating platform, mobile phones are the ideal environment to run the client software.

Developer Tools

Being one of the first computers manufactures to introduce graphical interfaces, the Apple macOS is intuitive and easy to use. As discussed earlier, macOS requires a purchased licence, and macOS is designed to work specifically on Apple's hardware. It does support Java software development tools.¹⁹ Furthermore, regarding *Apple iOS* app development, it is the ideal platform since Apple integrates their hardware and software together, and Apple has tools for developing iOS mobile applications.²⁰

¹⁶ See Apple (n.d.). About macOS Server 5.7.1 and later. Retrieved on Jun. 7, 2024 from: <https://support.apple.com/en-us/101601>

¹⁷ This is discussed on Java (n.d.). *How do I get Java for Mobile device?* Retrieved on May 22, 2024 from: https://www.java.com/en/download/help/java_mobile.html

¹⁸ There are many Linux distributions, so I cannot assess all of them in a reasonable time for app development and deployment. So, the best I can do is assess mainstream distributions, like the *Debian* and *CentOS* family of Linux distributions, conclude through experience that they come with standard web browsers that support HTTP protocol, REST API, and can parse JSON output.

¹⁹ Isah, J. (2023). *How to Set Up Your Java Development Environment*. freeCodeCamp. Retrieved on May 23, 2024 from: <https://www.freecodecamp.org/news/how-to-set-up-java-development-environment-a-comprehensive-guide/>

²⁰ Apple's XCode is a tool for developing iOS apps; see the following for more information (Retrieved on Jun. 7, 2024):

While Linux distributions are mostly free to use, their intuitiveness and IDE support depends on the distribution used. Some Linux distributions can run Java development tools.²¹ In particular, the Ubuntu Linux and other members of the Debian family of Linux distributions support various integrated development environments for developing *Java* and *Android* apps, such as *Eclipse IDE*²² and *Android Studio*.²³ Regarding the development of iOS mobile applications, the software development IDEs *Flutter*²⁴ and can be used to develop them.²⁵

Finally, Microsoft Windows desktop operating platforms are easy to use and the most common operating system in use today. Windows desktop operating platforms do require a licence to use, but it can run on different hardware platforms, and can run Java development tools.²⁶ *Visual Studio*²⁷ can be used to develop *iOS* and *Android* mobile applications.

While it may be possible to develop software on a mobile phone, it is not recommended, and I am not aware of any proper method to write and build software on the mobile phone platform.

Recommendations

Operating Platforms and Architecture

Regarding the service operating platform, I recommend using the Ubuntu Server Linux distribution because it is easy-to-use and free. *Ubuntu Server* also supports the *Java Runtime Environment*²⁸ and can run the server component that is *Draw It or Lose It*. I recommend using a cloud-driven operating platform, specifically *Amazon Web Services*,²⁹ since it is the cheapest and makes it easier to maintain such systems.³⁰

Regarding the client side of the software, it comes in the form of a web interface. Any operating platform that comes with a web browser that can render HTML5 content and execute JavaScript will be

https://developer.apple.com/library/archive/referencelibrary/GettingStarted/URL_Tools_for_iPhone_OS_Development/_index.html

²¹ Kumar, S. (Apr. 10, 2023). *Best Java IDE's for Linux Developers*. Tutorials Point. Retrieved on May 23, 2024 from: <https://www.tutorialspoint.com/best-java-ide-s-for-linux-developers>

²² Eclipse IDE website (Last Retrieved on Jun. 7, 2024):

<https://www.eclipse.org/downloads/packages/release/helios/m7/eclipse-ide-linux-developers>

²³ See the following article (Retrieved on Jun. 7, 2024): <https://developer.android.com/studio/install>

²⁴ Flutter (n.d.). Retrieved on Jun. 7, 2024 from: <https://flutter.dev/>

²⁵ I do not have much experience with Flutter, so I am not totally sure how reliable it is when it comes to developing iOS or Android mobile applications.

²⁶ See Isah (2023).

²⁷ Visual Studio (n.d.). *Mobile Application Development*. Retrieved on Jun. 7, 2024 from: <https://visualstudio.microsoft.com/vs/features/mobile-app-development/>

²⁸ Ubuntu (n.d.). *Install the Java Runtime Environment*. Retrieved on May 23, 2024, from: <https://ubuntu.com/tutorials/install-jre>

²⁹ From my research, it appears that *Amazon Web Services* is the cheapest in terms of VPS and dedicated server offerings. This will allow for easier scaling of hardware resources should *Draw It or Lose It*'s userbase grows.

³⁰ While cloud computing takes care of the infrastructure, and in some cases, the platform, systems engineers still need to plan and consider external factors like the expected size of the userbase, number of images to be rendered in the game, and how that would affect the cloud computing resources. Despite its "serverless" architecture, there is still an underlying hardware component that is not infinite, and adding resources will cost more money.

able to run the game. It can be assumed that the three major operating platforms, *Microsoft Windows*, *Apple macOS*, and the various *Linux* distributions, come with a standard web browser, and that the end-user is educated enough to use their web browser.

Finally, regarding the development tools, this will need to be looked at more. It will depend on the subjective preferences and tastes of the developer and other factors like the organisation's "bring your own device" policies, the preferred operating platform of the organisation, and others that did not spring to this author's mind. Nonetheless, if deciding to use a *Microsoft Windows* system to develop apps, I recommend using *Visual Studio* for both the iOS and Android mobile applications, and the Java server-side component. If using a macOS system, Xcode, Android Studio and Eclipse IDE can be used for developing *iOS* and *Android* mobile applications, and the *Java* server-side component. If the developers are using a *Linux distribution*, *Flutter* can be used to develop the *iOS* and *Android* mobile applications, and *Eclipse IDE* can be used to develop the Java server-side component.

Storage and Memory Management

Regarding storage, these operating platforms typically come with their own internal hard drive or flash memory. The server and developers should use some kind of backup solution to prevent losing programme files or user data regarding the software. The end-users with the web client do not have to worry about backups,³¹ as the server stores their information.

In a previous research attempt,³² I was given the following problem: assuming that *Draw It or Lose It* will store two hundred (200) images at eight megabytes (8 MB) per image, work out the necessary amount of data that the volatile non-volatile storage will need to store the game's programme files. To solve this, I developed a simple linear model to determine the minimum space needed by the software:

$$\mathbb{E}(\text{Space [in GB]}) = \beta_0 + \beta_{\text{game}} + \beta_{\text{images}} \cdot x_{\text{images}}$$

The beta coefficients are a scaling factor (in gigabytes) of a component of the game, and the "x" variables are the amount of a component (assuming that the component is additive and more-or-less the same size as its respective beta coefficient). The Beta_0 is the space taken up by the operating system and other auxiliary software (in gigabytes), the B_game is the amount of space taken up by the game software (in gigabytes), the Beta_images is the size of each image (in gigabytes), and the x_images is the number of images that the game will store.³³

Assuming that each image is eight megabytes (or about 0.008 gigabytes) and that there are 200 images, the estimated number of space needed to store the game is: **B_0 + B_game + 1.6[GB]**. More research

³¹ Specifically regarding the *Draw It or Lose It* game. Everyone should back up their data onto external mediums and even third-party backup services in case of some event that causes their computer systems to cease functioning.

³² Ahmann, A. (Jun. 15, 2024). *CS-230: Assignment 6-1: Module 6 Journal*.

³³ It should be noted that these beta coefficients are not "set in stone." Empirical and statistical techniques can be used to estimate the values of these beta coefficients, and we can treat this linear model like a random model, where each image may be smaller or larger than average. But a large enough images sample of will "cancel out" the outliers and the expected value of space taken up will, hopefully, be nearly equal to the actual space taken up.

will be needed to determine the values of B_0 and B_{game} so that a concrete number can be produced.³⁴ What can be said with near certainty is that the game will need at least 1.6 gigabytes to store images, and that we should round up to two gigabytes (2 GB) just to be safe.

Regarding memory management, typically these operating platforms come with their own *random access memory* chips. Developers need not worry much about memory management, as the Java runtime environment is designed to mitigate potential issues with memory leaks and other related problems.³⁵ As with estimating the amount of RAM needed, I created a linear model like the one that I presented to estimate the number of gigabytes of storage needed to store the server component programme files and operating system --- the major difference being that I am estimating how the number of users would affect the RAM usage.

Distributed Systems, Networking and Security

The server component must use an operating platform that is “fine-tuned” to function as a server. I recommend using *Ubuntu Server*, but any kind of server operating platform like *Red Hat* or *Windows Server* could be used if needed. A simple client-server model will be employed where multiple end-users connect to the *Draw It or Lose It* web service and the server Java-powered web service will manage instances of each `Game` and the `Teams` involved in them.

In terms of cybersecurity, the server component of the software should have a basic accounting system that allows end-users to only access data that they are authorised to. Passwords should be hashed and stored securely. A basic SSL/TLS encryption mechanism should be employed for data being transported from client-to-server, and vice versa. When the game is accessing files, a `FileLock` should be used³⁶ to prevent overwriting a file and maintain integrity.

Finally, third-party red teams³⁷ and/or exploit developers should be employed to audit the Java server component and identify potential vulnerabilities with the application. Developers should fix vulnerabilities when identified.

³⁴ The beta coefficients are just coefficients that act as placeholders for values that are (practically) constant. If we do not know their value, we can just represent them with a letter that acts as a placeholder for us to solve (either by the applying the of algebraic manipulation, or by the method of empirical observation).

³⁵ “Geeks4Geeks” (Dec. 14, 2018). *Java Memory Management*. Retrieved on May 23, 2024 from: <https://www.geeksforgeeks.org/java-memory-management/>

³⁶ See *Operating System Concepts* (Silberschatz et al. 2008, §10.1.2)

³⁷ “TechTarget Contributor” (c.a. Apr 2021). *What is red teaming?* Retrieved on May 23, 2024, from: <https://www.techtarget.com/whatis/definition/red-teaming>