

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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)4

[**Domain Model**](#_8h2ehzxfam4o)4

[**Evaluation**](#_2o15spng8stw)5

[**Recommendations**](#_m8aleynsvzvc)9

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 11/11/2020 | Vitalie Cucuta | Updated all information for the initial design. |
| 1.1 | 11/24/2020 | Vitalie Cucuta | Revised executive summary, revised domain model, and revised evaluation. |
| 1.2 | 12/09/2020 | Vitalie Cucuta | Revised recommendations section. |

**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 Gaming Room would like their current game, Draw It or Lose It, to be developed as a web-based game that is available on multiple platforms. This game is currently only available as an Android app only. Our company will develop this game as a web-based game as well as available on Android and iPhone marketplaces. Draw It or Lose It is a game loosely similar to a 1980s TV game Win, Lose or Draw, where teams compete to guess what is being drawn. The game will contain a library of stocked images to use for players to guess where each image will be rendered at a steady rate and fully complete at 30 seconds. If the guessing team does not guess the image within 15 seconds, remaining teams have an opportunity to offer one guess each and solve puzzle in the remaining 15 seconds. Design constraints for Draw It or Lose It include, but not limited to, multiplatform availability, user interface, image library stock, image licensing, and pre-existing intellectual property. The software requirements will be met across all platforms and the game will be the same no matter what platform it is played on.*

## [Design Constraints](#_2et92p0)

1. *Multiplatform availability (Technical Constraint) –* 
   1. *Due to this game being across all platforms, we will need to make sure that we develop this game in a format that can be distributed amongst all platforms.*
   2. *The programming language that we will need to use must be compatible with all platforms so that multiple languages don’t have to be used which would require more work when changes need to be done down the road.*
2. *User interface (Technical Constraint) –*
   1. *Since the game has already been released on the Android marketplace then the user interface and design of the game should stay the same.*
   2. *Upgrading the design would be a better option going forward because creating a new design may draw away from the success of the game so far and could cause users to not like the new direction of the game.*
3. *Image library stock (Technical Constraint) –* 
   1. *The game that is currently available uses stock library images already and this library and the image types must be compatible among all platforms.*
   2. *If the image type or library cannot be supported by one of the platforms, then it will cause the game to not work properly and the images will need to be transferred over to a valid framework or image type.*
4. *Image licensing (Business Constraint) –* 
   1. *The images used within the game must be either allowed to be used or the license for their use must be available.*
   2. *Using images that may cause copyright infringement may cause losses, be they monetary or business, to our company as well as yours so all documents showing allowed use of these images must be provided to us.*
5. *Pre-existing intellectual property (Business Constraint) –* 
   1. *This game is based on an old game show and although there are no issues with this game on the Android platform, we must make sure that there are no licensing restrictions due to this.*
   2. *As with the image licensing, we must make sure that there is no infringement upon intellectual property of the game show so that there is no loss on both our ends.*

## 

## [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 UML class diagram below is a static structure diagram that will show the classes, variables, and methods that will be used in the code development.*
* *At the top left of the diagram we have the ProgramDriver class with an arrow pointing to the SingletonTester class showing that the ProgramDriver class will use the SingletonTester class to test the code. Since the requirement of this game is to only have one instance where this game can exist in memory, we must use the Singleton method to create this restriction.*
* *The ProgramDriver will be the code that will create actions to occur within the game and will be the main source code that will run the program itself.*
* *The GameService class is going to hold the bulk of the source code and the more complex methods that will be the skeleton of the game itself. These functions will be used in the ProgramDriver and doing so will make the code much easier to read.*
* *This type of data structure is known as encapsulation. We have data in them GameService class, but it cannot be accessed directly and the only way to access it is using methods.*
* *The use of methods in coding lead to much cleaner source code and it is also much simpler to debug and fix the code down the road. It basically makes it not be a jumbled-up mess.*
* *The GameService class may look like it’s linked together with three other classes and that is essentially exactly what it is. It is linked to the Game, Team, and Player classes.*
* *As per your requirement, we will use these classes for having a unique game, team and player.*
* *The links between these four classes means that they are associated to each other.*
* *The GameService class can have zero or more Games associated with it.*
* *The Game class can have zero or more Teams associated with it.*
* *The Team class can have zero or more Players associated with it.*
* *The Game, Team, and Player class all diverge into the Entity class.*
* *The arrow used to show the divergence means that the Game, Team, and Player classes inherit the properties of the Entity class.*
* *All these classes work together to create the final product for what the end user will see when playing the game.*
* *This skeleton structure is what isn’t seen by the end user, but it is what makes a game run and it is the same across the board for other games. Although not all games use this same structure, it is the same idea.*
* *We will use this to create a final product that is exactly as you imagined it to be.*

****

## 

## [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 does offer a server-based deployment method, but it is in older versions of the OS. Mac OS servers are more reliable and easier to use if used to Mac software. The downside to using Mac based server web hosts is that these Mac servers are harder to find and cost more because they are in less demand.  Advantages:  +Unix OS  +Open Source Servers  +Secure  +User and file access controls  +Support LDAP and ADP  Disadvantages:  -Docker support virtual only  -Difficult to implement server-only environment | Linux is the most popular OS to be used for web hosting and it is also the OS used to run the servers of Google.  Advantages:  +Unix OS  +Open Source Servers  +Secure  +User and file access controls  +Support LDAP and ADP  +Docker support  +Large online/cloud support, internet runs on Linux  Disadvantages:  -Need to decide on vendor support: Redhat or Ubuntu | | Windows is a good all-around OS for web hosting, and it is the practical alternative to the other OS’s. Windows offers its own server-based deployment called Azure. Windows offers a lot of support and makes it easier to use for web hosting and because Windows is so well known, it makes hosting a lot easier. The biggest disadvantages are that Windows servers can require frequent reboots and Windows licensing is extremely expensive.  Advantages:  +Secure  +User and file access controls  +Support ADP  Disadvantages:  -Need to use server version  -Limited LDAP  -Poor open source server support | Oracle offers a database mobile architecture and using the Mobile Development Kit it offers tools for packaging, publishing and testing applications.  Advantages:  +iOS developer tools  +iOS multi-device support (iPad, iPhone, watch, TV)  +Android developer tools  Disadvantages:  -Android poor multi-device support |
| **Client Side** | Cost of software development: Fairly high cost due to limited use.  Expertise for development:  Moderate – High. Mac OS is much easier to use if familiarity is there.  Mac OS can require multiple third-party applications to be used for better development of a web application and these will be utilized for compatibility with all other platforms. Use of a web host tool that runs both on the server and locally means that the tool used is likely cross-platform and allows for cross-platform use.  Advantages:  +Support Safari browser  +Firefox  +Chrome  Disadvantages:  -No IE  -Small install of client desktops | Cost of software development:  Very low cost for development.  Expertise for development: High. Linux is open source but because of that requires much more knowledge of the operating system because there isn’t as much manufacturer support as there is with Mac and Windows.  Linux OS is highly compatible with all other platforms and can run smoothly on them so the versatility of Linux must be noted when considering this OS. Use of a web host tool that runs both on the server and locally means that the tool used is likely cross-platform and allows for cross-platform use.  Advantages:  +Support firefox  +Chrome  Disadvantages:  -No IE  -No Safari  -Small install of client desktops | | Cost of software development: Very high due to windows licensing requirements.  Expertise for development: Low - Moderate.  Windows is a very well-known OS so most will find it much easier to use and Microsoft offers support for their technologies. Use of a web host tool that runs both on the server and locally means that the tool used is likely cross-platform and allows for cross-platform use.  Advantages:  +Support large installed base of desktop clients  +Firefox  +Chrome  +IE  Disadvantages:  -No Safari | Mobile OS’s make it difficult to create a web-based application. The OS’s that exist work best for mobile applications rather than web-based application and this limits cross-platform availability without third-part software to run mobile OS applications.  Advantages:  +iOS: Firefox, Chrome, Safari  +Android: Firefox, Chrome, Brave |
| **Development Tools** | Tools Used:  Parallels Desktop, Local, SmartSVN, iTerm2, Shortcut Bar, Sequel Pro, Forklift, Keyboard Maestro.  Written in: C, C++, Objective-C, Swift, Python, Java, Ruby, AppleScript, JavaScript.  Advantages:  +iOS native and Android development supported  +High productivity environment  +Large language support including Swift and objective-C  +Java Support  +Unix shell scripting  Disadvantages:  -Small pool of developers  -Difficult/ impossible to test IE browsers  -Few IDEs, but high quality | | Tools Used: Node.js, XAMPP, Insomnia REST Client, Visual Studio, Brackets, Pinegrow Web Editor.  Written in: C, ASM, HTML, JavaScript, Python, Java.  Advantages:  +New technology first platform  +Large developer pool  +Largest set of IDEs with quality but no always support  +Large language support including Swift and objective-C  +.NET support  +Java support  +Unix shell scripting  +Powershell  Disadvantages:  -Android dev support only: native | Tools Used:  Visual Studio IDE, Visual Studio Code, Notepad++, Chrome DevTools, Visual Studio Browser Link, VS Code Debugging.  Written in: C, C++, Java, JavaScript, ASM, HTML, Python, CSS.  Advantages:  +Large set of IDEs with support and quality  +Large language support including C#  +.NET support  +Powershell  +Java support  Disadvantages:  -Android dev support only: native | Tools Used: Oracle Mobile Development Kit, Oracle Database Mobile Architecture, Visual Studio, Apache, WebLogic Server.  Written in: C, C++, Java, HTML.  You don’t really develop on a mobile platform. |

## 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**: We recommend that the OS you choose for this project is Linux. It is a very versatile and easily operated system using Unix-like OS, has open source servers, it is the most cost-efficient choice and it is a very secure operating system as well.
2. **Operating Systems Architectures**: The Linux operating system architecture can be broken down to 4 categories. These categories are the kernel, shell, hardware layer and utilities, such as system library and system utility. First is the hardware layer which consists of the peripherals so for a computer system that would be the HDD, RAM, CPU, etc. The kernel is the core of the OS and it is the responsible for all major activities of the OS, so it interacts with the hardware directly. The shell is essentially the face of the kernel, so it hides the complexity of the kernel and allows the user to input commands for the kernel to execute. The utilities include the system libraries which implement the functionality of the operating system and the system utility programs which are used for specialized, individual level tasks. (Operating System – Linux, n.d.)
3. **Storage Management**: With a web-based application, the main method of storing information will be through the cloud. Since we will be using Linux as the platform for this application, we can use Google Cloud for all storage needs and for a browser-based game, especially since Google also runs on Linux. Google also provides their own security features for their Cloud.
4. **Memory Management**: The language in which this application is being written in is Java and the great thing about Java is that it manages its own memory and doesn’t need any explicit intervention to allocate and deallocate memory such as one would in C language, for example (GeeksforGeeks, 2018). When there are memory leaks or memory overconsumption, it causes an application to continually slow down and eventually crash. With Draw It or Lose It, the game needs to always render and display pictures at a fixed rapid rate so if there are any kind of memory problems, it can cause slowdowns and the 30-second limit until the picture is displayed can take longer and ruin the entire experience. In Java, memory is handled by the JVM, Java Virtual Machine, but we cannot only trust that when it comes to a web application. JVM triggers the garbage collector process but because the garbage collection process causes the rest of the processes or threads to be paused, programmers must go through the process of applying algorithms, often termed Garbage Collector Tuning, to improve the performance of the program (GeeksforGeeks, 2018). Doing so will help with memory management in this game and keep the game from slowing down or crashing and generally makes for a much better user experience.
5. **Distributed Systems and Networks**: From the evaluation in the previous section, Google Chrome is available on all platforms and using this to play Draw It or Lose It will make it available across all platforms and can be connected through any device of choice. We can have one user on each type of OS, and they can still all play together using the Chrome browser. As with any web-based application, the users must have a good internet connection to play the game.
6. **Security**: All user information that will be stored in the Google Cloud will be protected with the use of the best security features which Google has to offer. Google constantly updates their browser security and they provide many security services such as Identity-Aware Proxy and encryption by default. Linux OS is also, by design, a more secure OS and because it is constantly managed by user-developers across the globe, it is extremely secure. Out of all the operating systems mentioned, Linux is definitely the most secure out of all of them.

Work Cited

GeeksforGeeks. (2018, December 14). Java Memory Management. Retrieved December 09, 2020, from https://www.geeksforgeeks.org/java-memory-management/

Operating System - Linux. (n.d.). Retrieved December 09, 2020, from https://www.tutorialspoint.com/operating\_system/os\_linux.htm