Draw It or Lose It

**CS 230 Project Software Design Template**

Version 1.0

Table of Contents

CS 230 Project Software Design Template……………………………………………………....1

Table of Contents………………………………………………………………………………....2

Document Revision History……………………………………………………………………....2

Executive Summary………………………………………………………………………………3

Design Constraints………………………………………………………………………………...3

Domain Model…………………………………………………………………………………….4

Evaluation…………………………………………………………………………………………5

Recommendations…………………………………………………………………………………7

Sources…………………………………………………………………………………………….9

Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 9/22/2023 | Rachel Siminski | Document created |
| 2.0 | 10/3/2023 | Rachel Siminski | Updated |

Executive Summary

Draw It or Lose It is a mobile game application produced by The Gaming Room. The aim is to create a web-based version of the game. The Gaming Room will need help transitioning from an Android app to a computer game.

Design Constraints

The game should be able to have one or more teams playing and each team should have multiple players. Each game should be four rounds long and each round should last for one minute. Within the minute round the computer should generate a new image which should be complete after thirty seconds. Each game and team will have unique names and users will be allowed to check if a name is already in use while choosing the name. Only one instance of the game should exist in the memory at any given time.

Domain Model



In the above UML diagram shows that the ProgramDriver (main class) uses a SingletonTester class to ensure the program is operating appropriately. The Singleton pattern ensures that there is only a single instance of an object with global access allowed to this object. Next, there are the classes for GameService, Game, Team and Player which are connected via 0 to Many relationships. This means that a Team may contain any number of Players, from 0 to Many. The same goes for each Game which can include any number of Teams, from 0 to Many. Additionally, the classes, Game, Team, and Player inherit from the superclass Entity, which acts as the foundational definition for each Entity. The attributes id and name are used with slight further definition (Player is composed of id and name attributes) in each of these three classes.

Evaluation

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac has powerful and efficient processing and the pre-installed apps don’t draw as much power as in Windows. The GUI is fairly the same across each Apple product. Mac can run Windows programs. | Less professional programs are compatible with Linux. While it is considered by most to be the best operating system, it is less popular in business environments. Less likely for cyber security attacks. Uses a command line | Uses graphical interface, making it more accessible to all users. Supports many popular programs. Very prone to user errors; integrated interface seen as a potential point of attack | Mobile devices include phones and tablets. Most if not all mobile devices are able to connect with the cloud and run apps. Tablets offer more versatility by allowing the use of additional hardware like keyboards. |
| **Client Side** | Mac products are very expensive and are less customizable. Newer Macs come with less hardware options, like less ports for plugging other hardware. Mac products may last longer than others. | Linux offers the most freedom, but this makes it less beginner friendly. Open-source and free to use. Long term support may not be available. | Software and hardware are more affordable than Mac. Beginner friendly. | Security risk through user error and through apps that may not be secure. |
| **Development Tools** | Swift is a programming language created by Apple specifically for programming various Apple products. Python, Ruby and C# are a few other options, but really the list goes on. | PHP, Perl, Python, Ruby, or MySQL. | ASP.NET, SharePoint and Exchange, VBScript | Java, Kotlin, C++, C#, Python, Swift, and more. |

Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

**Operating Platform**: Linux may be the best option for hosting the game server as it has the most freedoms available to the user, and the lowest cost to develop and maintain. Linux may also offer the most benefits in the way of performance in the game. However, Windows is a more common choice for business due ito its low cost and beginner friendly presentation. Windows also offers a much larger wealth of support in the event of any issues. Windows is also compatible with most professional programs. I would recommend Windows as the operation platform.

**Operating Systems Architectures**: With the recommended operating platform being Windows, the recommended architecture is Hybrid Kernel, which is a mix of a microkernel and a monolithic kernel. A microkernel is typically just a core and a number of plugins, which contains only what is needed to implement an operating system. A monolithic kernel contains everything inside the kernel, including memory management, at the user-level. A monolithic kernel on its own may provide high performance but can be less secure and less flexible when debugging. A microkernel is more flexible, since it is separated into the core and the plugins, but does not provide enough of a foundation on its own. The combination of these two architectures within the Hybrid Kernel structure allows for maximum flexibility, security and performance.

**Storage Management**: It may be best to utilize cloud storage as you have the ability to choose the size of the storage space, similar to purchasing a hardrive, and the cloud has the added benefit of not requiring physical space.

**Memory Management**: Windows uses RAM mostly but the Kernel-Mode Memory Manager dynamically and virtually allocates and reallocates memory, and divides memory into separate user and system address spaces. It would be helpful to utilize cache space in users’ browsers to pre-cache the images to be used in the game, allowing them to load faster.

**Distributed Systems and Networks**: In order to facilitate communication between various platforms it will be important to utilize an Application Programming Interface (API). This would define how the various components would interact and exchange data securely.

**Security**: Draw It or Lose It should utilize passwords and user authentication for each player when creating an account or accessing the game. There should be a secure API in place and it should be able to properly define the interactions between components. Encryption may also be useful as it ensures that even if data is stolen or leaked that the information is unreadable. The Gaming Room should also perform regular maintenance and security checks to ensure that the system is operating appropriately and securely.

Sources:

*12 best programming languages to learn on macos*. Lion Blogger Tech. (2023, March 21). https://www.lionbloggertech.com/12-best-programming-languages-to-learn-on-macos/

*Educative answers - trusted answers to developer questions*. Educative. (n.d.). https://www.educative.io/answers/what-is-windows-kernel

Exoft. (2022, July 5). *Mobile App Architecture: Best Practices*. https://exoft.net/mobile-app-architecture/

GeeksforGeeks. (2022, August 2). *Top programming languages for Android App Development*. GeeksforGeeks. https://www.geeksforgeeks.org/top-programming-languages-for-android-app-development/

GeeksforGeeks. (2023, May 12). *Monolithic architecture*. GeeksforGeeks. https://www.geeksforgeeks.org/monolithic-architecture/

Heusser, M. (2020, March 10). *What is a microkernel architecture, and is it right for you?: TechTarget*. App Architecture. https://www.techtarget.com/searchapparchitecture/tip/What-is-a-microkernel-architecture-and-is-it-right-for-you

*Hybrid kernel*. Techopedia. (1970, August 22). https://www.techopedia.com/definition/27004/hybrid-kernel

Inc., A. (n.d.). *Swift*. Apple Developer. https://www.apple.com/swift/#:~:text=Swift%20is%20a%20robust%20and,idea%20can%20create%20something%20incredible.

Learn how using a solid state drive (SSD) for gaming can result in faster load times and an improved gaming experience.1 2 Learn how using a solid state drive (SSD) for gaming can . (n.d.). *How ssds impact gaming*. Intel. https://www.intel.com/content/www/us/en/gaming/resources/ssd-improves-gaming-experience.html#:~:text=SSDs%20work%20without%20moving%20parts,on%20the%20potential%20of%20SSDs.

Nishadha. (2022, November 25). *UML class diagram relationships explained with examples*. Creately. https://creately.com/guides/class-diagram-relationships/#:~:text=The%20notation%200..,means%20%E2%80%9Czero%20to%20many%E2%80%9D.

Randi Wilson, L. L. (2022, April 21). *Simplify testing with the singleton pattern*. Salesforce Engineering Blog. https://engineering.salesforce.com/simplify-testing-with-the-singleton-pattern-1a53ba5c2c50/#:~:text=The%20Singleton%20is%20a%20creational,point%20of%20access%20to%20it.

Rehman, J. (2021, November 4). *What are advantages and disadvantages of macos*. IT Release. https://www.itrelease.com/2019/10/what-are-advantages-and-disadvantages-of-macos/

Sheldon, R. (2019, September 23). *Advantages and disadvantages of mobile devices in business: TechTarget*. Mobile Computing. https://www.techtarget.com/searchmobilecomputing/feature/Discover-the-benefits-of-mobile-devices-in-the-enterprise

Tedhudek. (n.d.-a). *Memory management for Windows Drivers - Windows Drivers*. Windows drivers | Microsoft Learn. https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/managing-memory-for-drivers

Tedhudek. (n.d.-b). *Windows kernel-mode memory manager - windows drivers*. Windows drivers | Microsoft Learn. https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-memory-manager

*What is encryption? how it works + types of encryption – Norton*. United States. (n.d.). https://us.norton.com/blog/privacy/what-is-encryption