

Draw it or Lose it

**CS 230 Project Software Design Template**

Version 1.0

**Table of Contents**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 05/24/24 | Alexandria Palm | Added the executive summary, design constraint, and domain model |

**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**

The major problems with creating a game application in a web based distributed environment are memory for holding all instances of the games, and commution between these instances in real time. We need to reduce loading times for individual objects as they need to be rendered for all players simultaneously. We also have to ensure that we can host enough games on a server at time to keep waiting times low for joining and creating a new game service. This all has to work through multiple operating systems and devices.

**Requirements**

The requirements that need to be met are working with multiple operating systems, server side memory, multiple device architecture, and bandwidth.

**Design Constraints**

The design constraints of building a game application to run on multiple types of devices housing different operating systems and communicating in real time over the web include, the operating systems and devices themselves, as well as, the bandwidth and size of the servers. The servers being small could negatively impact users when trying to create a new game instance. The devices need to be able to render the game instance and have it run smooth on thier respective operating systems. The bandwidth needs to be large on our end to seemlessly communicate with all devices on the instance so everything renders in real time.

**System Architecture View**

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**

The ProgramDriver class has the SingletonTester class as a test case to ensure functionality of the singleton instances that are going to be created using the GameService class. The GameService class has constructor to create a game instance, the Game class includes a constructor to create a Team instance, and the Team class has a constructor to create a Player instance. This is the Inheritance principle in which Player inherets properties from Team which in turn inherets properties from Game which inherets properties from GameInstance. These classes can create multiple unique instances of the lower classes on the inheritance tree. The Entity class houses all of the information from the Team, Tame, and Player classes to create a singleton instance of the game itself.



**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** |
| **Server Side** | Mac products are not well known for running servers. They may have the capability to run a server but there are other platforms more suited for web hosting. | Linux is very versatile in that it comes in many different forms. Most versions of linux are customizeable and include tools and packages that aid in server development. It is therefor more suited to host a web server. Linux is generally free, however there are paid versions that come with support contracts such as RedHat. | Windows is a prepacked OS that never stops improving on its capabilities. Microsoft has more recently been trying to expand its business to include server hosting tools such as Azure. Windows is a pricier option that requires paid licenses. | Mobile devices are the least suitable for server side hosting in that they rarely have enough physical hardware capacity to host a web based server and thus are not designed to do so. Because of this, you would never see a mobile device hosting a server in a data center. |
| **Client Side** | Client side development of browser based applications is primarily platform agnostic.  User testers accesssing the application on mac will want to ensure that all browsers available on Mac OS will be able to run the application. The application needs to be developed to run in a broswer (HTML,Javascript,etc) before testing. | There are many different varieties of the linux operating system, however they all generally have access to the same browsers as windows and mac operating systems. The testing in this environment will have to include all of these browsers. | There is a need to test the application on all web browsers supported by Microsoft, including Microsoft Edge. | There is already an andriod application and so porting to IOS will be less time consuming since there is already mobile capabilities in the application. |
| **Development Tools** | The relevant programming languages for developing a web based application would generally be Javascript, HTML, Python, Java, etc. This is true for Mac, Linux, and Windows. The application can be developed on any integrated development environment that supports browser based development. | Linux is a good development environment for web based applications due to its ease of use with open source tools and libraries. Testing can be great in a Linux environment because it has good support for virtual machines and containers to emulate other environments. | Windows offers visual studio and vscode which are some of the best development environments for a wide variety of applications and languages. | It is easiest to develop mobile applications from a desktop computer. The best IDE for developing mobile applications for Android, is Android Studio and the best for IOS, is Xcode. These have been created specifically for development on thier respective platforms. |

**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**: I would recommend using a Linux operating system for all development and server side deployment. Linux is very useful for developers because it offers a wide variety of tools for both development and server deployment.
* **Operating Systems Architectures**: Linux is the kernel for a fully customizeable operating platform that can be tailored to specific use cases. Linux supports a variety of software stacks. One example is the LAMP (Linux,Apache,MySQL,PHP) architecture.
* **Storage Management**: Storage management is an important task for a web based application. I would use physicasl hardware to better protect the data being stored by the application so cloud data breaches do not affect the users data.
* **Memory Management**: Linux seperates memory by usefulness, it allows the RAM to be utilized by the current running application. This will allocate space for the application that is running rather than pouring through irrelevant data.
* **Distributed Systems and Networks**: Having physical servers allows you to have a backup in case something goes wrong. The possibility of having a backup server allows for continued operation during times when one server experiences security breaches or power and functionality issues.
* **Security**: Linux is known to be a very safe OS. It is virtually immune to viruses since it is built to be as safe as possible. This protects the server for functionality and the users data. Redhat also has its own security team to help with this feature.