Draw It or Lose It

CS 230 Project Software Design Template

Version 3.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
3.0	10/16/2022	Ray Cooke	Third Edition

Executive Summary

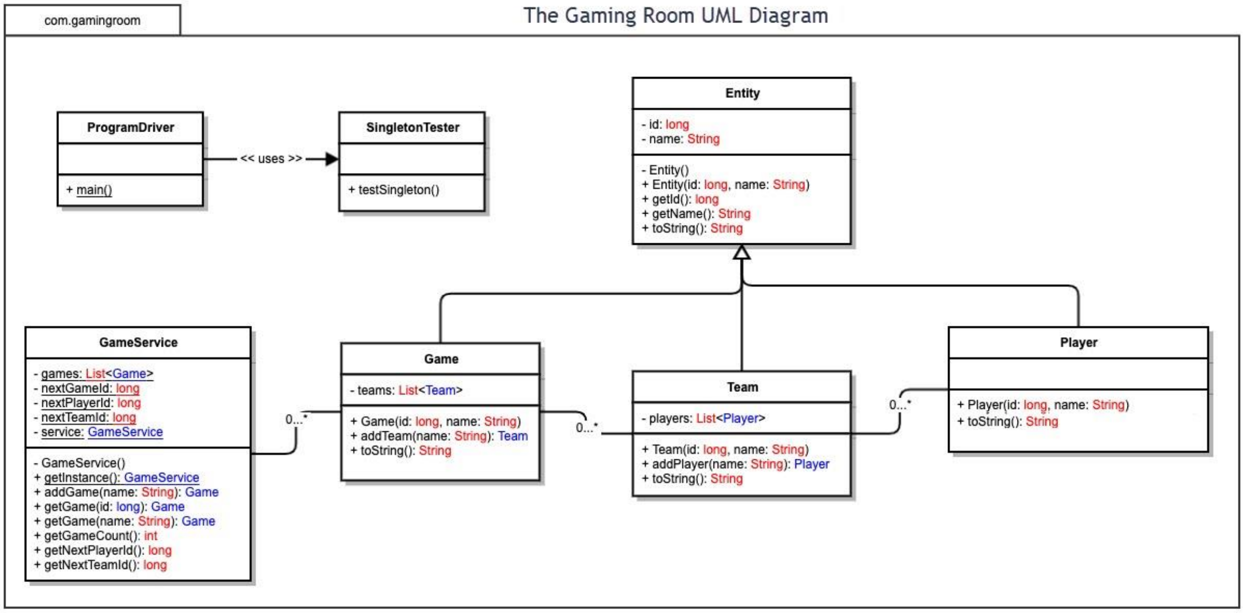
The Gaming Room requires assistance with getting their web-based interface established. They need a multiplayer game that groups unique teams and players together, no duplicates allowed. Only one game instance is allowed at a given time. Initial program logic will be written in Java.

Design Constraints

- Must isolate identifiers within ‘Game’, ‘Team’, and ‘Player’ classes
- Must use Singleton pattern in ‘gameService’ in ‘GameService’ class method to satisfy single game instance rule
- Must overload constructors in ‘Game’, ‘Team’, and ‘Player’ classes to enable multiplayer functionality.
- Project must be released in a web-based distributed environment
- Project is based on Android app

Domain Model

‘GameService’, ‘Game’, ‘Team’, and ‘Player’ classes are linked by zero to many relationships. This means that zero or more objects are related between these classes. ‘Game’, ‘Team’, and ‘Player’ are child classes of the ‘Entity’ parent class. This inheritance structure will reduce redundant code by allowing ‘Entity’ to store names and ids while the child classes organize the information. The main class named ‘ProgramDriver’ calls ‘SingletonTester’ upon execution of the program to verify only one instance of ‘Game’ is in memory.



Evaluation

Development Requirements	Mac	Linux	Windows	Mobile Devices
Server Side	Mac offers an easy-to-use server-side solution. Apple offers good customer support but offers a limited range of features. A downside includes the expensive proprietary hardware that must be acquired.	Linux is the most economic option. Linux provides open source, secure, and stable server-side solutions. A downside to Linux is that less support is offered by most distributers.	Windows is proprietary software that runs on a wide range of hardware. Microsoft offers strong customer support and security for Windows. A downside is the initial cost of the software.	Mobile devices have more weaknesses than advantages regarding their development characteristics. They lack the processing power to complete these tasks.
Client Side	Developing the game for Mac would require developers familiar with Swift. This would increase the level of expertise needed, cost, and time for the project. However, these efforts would also be compatible with IOS.	Developing and releasing to the Linux community would require a higher lever of expertise. This would also increase the time and cost of the development lifecycle.	Windows runs a wide range of programs, and the Windows gaming community is large. Developing the game on Windows would take less time, cost less, and require the least expertise.	Mobile devices are popular client-side systems. IOS and Android are popular platforms. It would take more time and money to develop two optimized applications. The different languages and IDE's involved would require a higher expertise level.

Development Tools	Swift is the leading programming language for Apple platforms, including Mac. XCode is a useful IDE for this environment. Code editors like Atom are also useful.	Eclipse and Apache NetBeans are useful IDE's for Linux. Many code editors are also available. The Linux Kernel is based on C and the Shell enables interaction with the Kernel. C++, Java, and Python run on Linux.	Some popular IDE's include Visual Studio and Eclipse. Many other code editors are available. Programming languages include C, C++, Java, R, and Python.	Swift is the leading programming language for Apple platforms, including IOS. The same development tools as Mac are sufficient. Java and Kotlin are the most used programming languages for Android. Android Studio and Eclipse are popular IDE's.

Recommendations

1. **Operating Platform**:

   I recommend Linux as the appropriate operating system to allow the gaming room to expand Draw It or Lose It to other computing environments. Linux is a popular operating platform for webhosting and has many exciting features for its benefit. Linux is also suitable for web development, the client side of the game will be developed in HTML, CSS, and JavaScript. For these reasons, Linux would be best suited for development of the server and client-side of the application.

2. **Operating Systems Architectures**:

   Linux is designed with the principal of least access in mind.
   The Linux kernel enables system hardware to complete computing tasks, it has the most access. The Linux shell acts like a liaison between the user and kernel, programs are run through the shell. Linux system libraries are functions that enable functionality while having less access to the kernel. Linux system utility programs are also run through the shell.

3. **Storage Management**:

   Linux provides native storage management solutions. A popular Linux distribution "Ubuntu Server" provides an option to encrypt the information that is being stored at the kernel level, partition drives, and provides RAID configuration abilities that mitigate the risk of a single storage drive failing. These features are typical on most Linux distributions and satisfy the requirements of the project.

4. **Memory Management**:

   Linux uses virtual memory; this memory management strategy keeps the amount of information in physical memory to a minimum while maintaining the principal of least access. The kernel controls memory allocation, it achieves this by mapping, storing to hard disk, and pointing towards hierarchal tables of information that is in memory. Using these memory management protocols will ensure smooth operations and speedy load times for images within the game.

5. **Distributed Systems and Networks**:

   Draw It or Lose It will be distributed to multiple software interfaces and connected by a server. As the game scales up, more servers should be added to handle the increased load. This will accommodate an influx of users and perhaps more images if The Gaming Room desires them. A REST-style API will be used to define interactions between the client and server side of the game. RESTful APIs are a proven method to facilitate the wants and needs of the client. They provide a flexible method for uniting the client and server side of an application. The multi-user nature of Draw It or Lose It requires the server to be running to enable gameplay. To mitigate downtime, user analytics should be observed, and maintenance outages should be planned when minimal users are playing the game.

6. **Security**:

   To protect user information and system integrity, granting the least amount of access necessary for users is an essential practice. The database of users and teams should be protected and only available to administrative users. “Dropwizard” will be added to the RESTful API to authenticate and manage users. User regulation and query abilities will also be available for administrators, this will help to detect and eliminate bad actors using the service. Linux is a secure medium for server-side development and maintenance mostly due to its open-source nature, and compartmentalizing processes and user privileges. Many distributions are shipped with native firewall functionality. Using a firewall will increase the security of the server. Preventing social engineering is a physical step that The Gaming Room should take very seriously to protect their assets.

References

Architecture of linux - javatpoint. www.javatpoint.com. (n.d.). Retrieved October 15, 2022, from https://www.javatpoint.com/architecture-of-linux

By: IBM Cloud Education. (n.d.). *Rest-apis*. IBM. Retrieved October 15, 2022, from https://www.ibm.com/cloud/learn/rest-apis

Concepts overview. Concepts overview - The Linux Kernel documentation. (n.d.). Retrieved October 15, 2022, from https://www.kernel.org/doc/html/latest/admin-guide/mm/concepts.html

Configuring storage in the server installer. Ubuntu. (n.d.). Retrieved October 15, 2022, from https://ubuntu.com/server/docs/install/storage

Day, B. (n.d.). *How secure is linux?* Linux Security. Retrieved October 15, 2022, from https://linuxsecurity.com/features/how-secure-is-linux

Linux architecture. TecAdmin. (2020, October 29). Retrieved October 15, 2022, from https://tecadmin.net/tutorial/linux/linux-architecture/