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	11/12/21	Kathryn Snow	Add working game, team and player classes in requested software design with unique game identification.

Executive Summary

We will create an online game which allows teams of multiple players to choose a unique name. Multiple teams may play in a game. The games can be saved by an identifier for each instance of a game, team, or player. This will require a class for game, team, and player have a relationship with a new class called 'Entity', so they share common attributes. We will make it so 'team' has a relationship with 'player'. 'Game' has a relationship with 'team' and 'GameService' has a relationship with 'game'.

Design Constraints

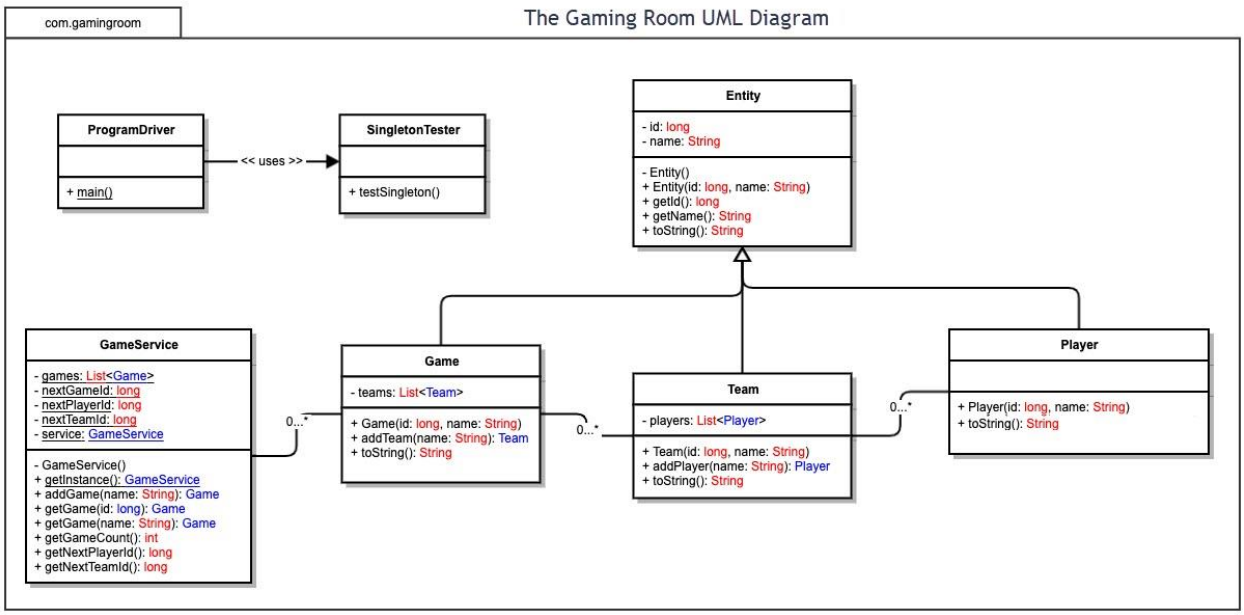
Android, iOS, and web-based access requires different software development. This means more people or longer development time for only one team. With each platform more than one user team will be accessing the game at the same time so each platform must work smoothly together. This is a consideration for the development team(s). Beginning the game there needs to be a team leader (first person) to choose a unique name with a method to allow others to join. This is an example of how each platform needs to work smoothly. Each game with its team members needs a unique Id for each instance. This also needs to work smoothly overall between the platforms.

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

Game, Team, and Player, each have a relationship with Entity where they are an Entity. This is called the "is a" relationship. It means they are inherited from Entity. Like children they have common attributes, id and name, and Entity is defined as a Super class. Team and Player has a relationship where Player is a part of Team. Team and Game has a relationship where Team is a part of Game. The same is true for Game and GameService, Game is part of GameService. The ProgramDriver uses a SingletonTest to insure easier implementation, change, and test of objects.



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	Good graphical user interface offering ease of use. It has quality hardware tuned to the operating system. It has tight integration with iPhones and iPads. Mac is limited to only what Mac allows.	Known for versatility and embedded solutions. It is an open-source operating system that can be modified and extended by anyone. Anyone with the skills can work on Linux.	Windows have many options in hardware. This allows for inconsistencies in functionality and quality. Malware, spyware, and ransomware aim more toward Windows.	Mobile Devices are varied in their specifications and compatibility. They also have the poorest security.

Client Side	A Mac user can use any version of Mac because of the consistency. For users, Mac is expensive but the integration between Apple products is tight	Linux is not a commonly used system because it is not standardly placed on any computer like Microsoft or Apple. It could take some time to become comfortable with.	Windows has changed dramatically during different versions. It is the most used system in the world. It is more likely than either Mac or Linux to have security risks.	Users love the flexibility of mobile phones but are often frustrated by the limits they experience between a computer and a phone.
Development Tools	IDE most recommended for Mac System is Xcode which uses JavaScript.	IDE most recommended for Linux System is NetBeans and Eclipse among others with C++ suggested. JavaScript has an install (Node Js runtime) to work on Linux.	IDE most recommended for Windows System is Visual Studios. Eclipse and NetBeans are also useful. JavaScript is used among others.	Using Android Studio for Android mobile phones and Xcode for iPhones. You can use JavaScript for both IDEs.

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**: I recommend Windows operating system to run The Gaming Room based on the larger user base and flexibility. This allows for a larger selection of software programs, and it is preloaded with many productivity tools. It also is backwards compatible, meaning old programs will run on newer versions. Using Windows will allow "Draw It or Lose It" to reach the largest customer base and the user to experience a satisfying experience with the product.

2. **Operating Systems Architectures**: Windows operating system offers many software packages and a wide variety of programming options. Our game benefits from applications like a Graphical User Interface (GUI) and PowerShell configuration used to manage server roles. Windows OS also supports many programming languages which provides more flexibility throughout the life of the game.

3. **Storage Management**: Windows server operating system has wide range of storage configurations, from single disk desktops to external storage arrays. You can manage files on your hard drive with information of the file location and size plus use cloud storage. Storage subsystem manufacturers can also support Window based storage management in their products.

4. **Memory Management**: Windows has memory management options that include virtual and physical address space using two and four gigabytes of memory. Microsoft documentation page explains that "each process on 64-bit Windows has a virtual address space of 8 terabytes. All threads of a process can access its virtual address space. However, threads cannot access memory that belongs to another process, which protects a process from being corrupted by another process." This and more make up the useful way Windows is recommended for use in this proposal.

5. **Distributed Systems and Networks**: Windows uses cloud computing which is splitting software systems into multiple components with each component running on a different computer and performing a specialized function. The client, as a node, is one of many nodes which reach out to others with an input and wait at their station for output. It is vast and there can be some issues with routing and congestion. It is amazingly reliable considering the opportunities for failure.

6. **Security**: Microsoft describes security as "Windows Server Security provides layers of protection built into the operating system to safeguard against security breaches, help block malicious attacks, and enhance the security of your virtual machines, applications, and data." They have anti-spyware built-in and VPN, virtual private network, capabilities. Windows use client RDP (remote desktop protocol) which is when the user has graphical interface connect to another computer over network connection. This step could be a weak link for cyber criminals to use. Always use an additional security method to back up the built-in method.