



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
Requirements	3
Design Constraints	3
System Architecture View	3
Domain Model	3
Evaluation	4
Recommendations	5

Document Revision History

Version	Date	Author	Comments
1.0	08/05/2023	Andrew Obrochta	Project Two

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 Gaming Room requested that CTS provide a web-based version of their Android-based game Draw It or Lose It. The game ought to support numerous teams, each with several players. Only one of each team, player, or game-instance should be present.

To prevent numerous game instances, a singleton creation pattern has been adopted for objects, and an iterator approach will stop conflicting teams and team members.

Requirements

Multiple teams must be able to participate in a game.

There may be numerous players on each team.

Every team and name must be distinct, and the program must look for overlaps.

The game must be created using a singleton design, in which only one game can be active at once.

Design Constraints

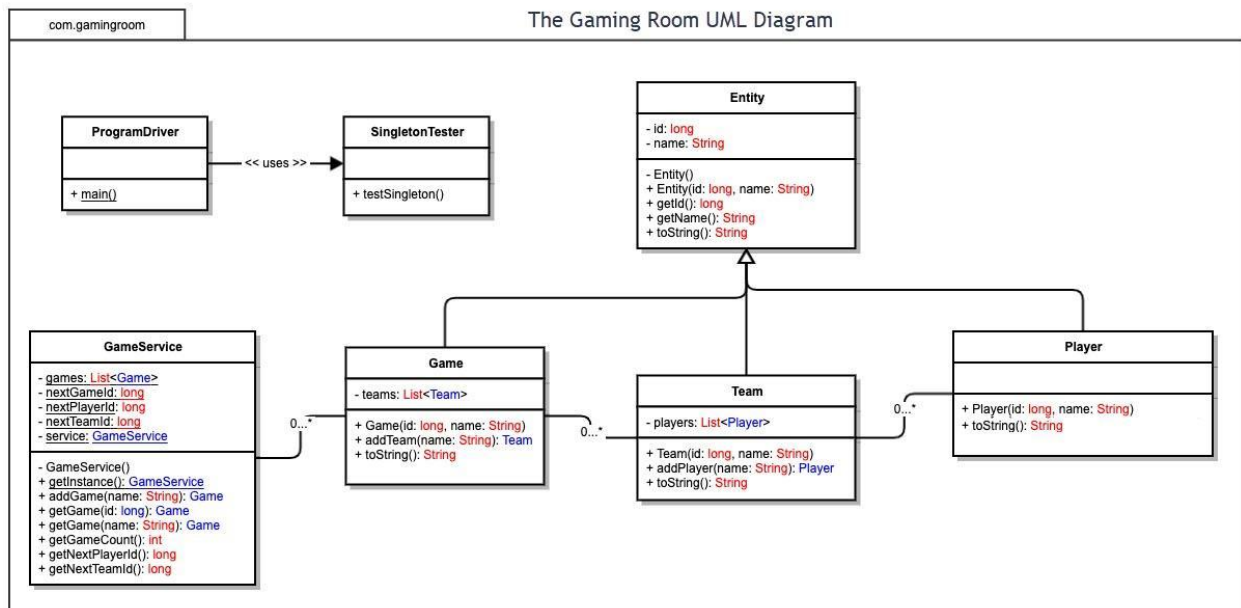
An Android-based deployment for The Draw It or Lose It is now in use at The Gaming Room. The tech must support web-deployment because CTS has been asked to expand this on the web. Java has been chosen for this reason. Java, the programming language of the Android SDK, should make this new deployment attempt simpler. It is necessary to review and/or improve the Android platform's current APIs for mobile use.

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 games, teams, and players will be created using a key driving course from the program. Due to the singleton design approach employed in the GameService class, which actually performs the building, only one GameService class may ever exist in memory at any given time. GameService stops the creation of any instances by setting its constructor to private. The only way to construct a GameService is by utilizing the getInstance method. GetInstance determines whether GameService has been started, and GameService is only started if it hasn't already been loaded into memory. Once GameService is operational, the driver class can call the function addGame. AddGame uses the iterator pattern to prevent the creation of Game objects bearing the same name. This new Game object is then added to the List of games. Once a team has been built, it may be added to the game using the addTeam method. AddTeam employs the iterator pattern to prevent the game from receiving more Team objects with the same name. This new Team object is then included in the List of teams. After the team has been built, a player may be added by using the addPlayer method. In order to prevent other Player objects with similar names from being added to the team, the iterator pattern is utilized in addPlayer. This new Player object is then included in the list of players.



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	Although OS X Server is available for Mac, finding hosts will be challenging and expensive unless the client wants to buy their own hardware.	The most widely used web hosting OS is Linux. Open Source operating systems like Linux typically have lower maintenance and license expenses than closed operating systems like Windows.	Windows servers are convenient because they are GUI based and many office apps may run on the server, providing plenty of comfort. In contrast to Linux, license charges are frequently very costly per user.	However, they are not designed for multi-user serving. Mobile devices can be used as personal web servers or file servers. They are not scalable like blade servers and often have more constrained hardware, such as RAM.
Client Side	The consumer is compelled to purchase an Apple-built MAC system since it is more expensive than Windows.	Most people agree that Linux is among the best operating systems available. Particularly true for software developers. Only cost is getting a linux operating computer.	Windows is often written using the widely used programming languages C# or.NET. The creation of a Windows client program would not present any obstacles.	Multi-user functionality is not intended for mobile devices. However, creating a client application for iOS or Android is simple. Because the Android SDK is built on Java, code created for Windows and Linux may be used as a starting point. The hardware requirements for Mac also apply because iOS is SWIFT-based.

Development Tools	Objective-C and SWIFT are the development languages used by Mac. Xcode is also used but is somewhat pricey.	Linux development can be done in Python, Java, or C/C++. Python IDEs, like NotePad++, are frequently free. An additional well-known Python IDE is PyCharm.	C# and .NET are primarily used in the development of Windows. Microsoft's Visual Studio is a very well-liked IDE that provides a variety of plugins.	<p>The most popular Android IDE is Android Studio, which was created by Google as the official development tool. The Android SDK is Java-based. Downloading Android Studio is cost-free.</p> <p>The Objective-C and SWIFT programming languages for iOS are virtually entirely created in XCode.</p>
--------------------------	---	--	--	--

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:** The best programming languages and development environments for building this kind of applications will depend on the platform and operating system. To ensure both the pace and quality of the job, I suggest having two teams, one that focuses on mobile platforms and the other that concentrates on desktop computer platforms. Within these teams, there will be divisions, with a team each for the iOS and Android operating systems in the case of mobile. Since the game may be played online or through a browser on any desktop operating system, the Desktop team won't need to be further divided.
2. **Operating Systems Architectures:** Different languages and silicon are used for each operating system. Most modern hardware is coded in C or C++, but because of the devices' unique user interfaces, their software is separate. Mac OS and Windows offer different user interfaces and experiences because they place different priorities on those areas. This also results in market competition, which is good because it encourages innovation. Due to the modifications to the user experience, applications must function in accordance with that OS. The reasons why some software is only available in particular operating systems include some of the needs that Mac OS has in comparison to Windows, and vice versa.
3. **Storage Management:** Because it influences how much data can be stored and accessible promptly, storage management is essential. Despite the outstanding quality of the consumer and business hardware currently available, it is still a good idea to store Draw It or Lose It data in a cloud storage service. If the data is kept in a third-party data center, you no longer need to maintain your own data server for a certain application or piece of software. When compared to maintaining your own server, the overall cost will be lower.
4. **Memory Management:** The variables and specific tactics needed to ensure memory is handled appropriately in the Draw it or Lose It software include storage type, speed, consistency, and security. Keeping high-definition image files demands a lot of memory and storage space because each high-definition image file is a specific number of megabytes in size and there are a certain number of total gigabytes of storage space. Furthermore, the appropriate technology must be available to store and send the data files without compromising precision, speed, or image quality. For the game to run smoothly and have instant pop-ups, the visuals must be sent at a specific MB/s.
5. **Distributed Systems and Networks:** Since they ensure that data may be transmitted and used on all kinds of devices, distributed systems and networks are essential for connecting customers utilizing diverse devices. Because several operating systems and device types exist, it is sometimes necessary to convert files and data from one to the other because they are incompatible. If the game's data is kept on a server, the Draw It or Lose It game can be accessible by multiple devices and device categories.
6. **Security:** Although every application has varying amounts of protection, protection is essential. The first line of defense consists of the coding processes used by the development team. The

next line of security defense is creating real security programs that can detect and halt attacks. The final line of defense is provided by the security features built into the servers' and hardware. There is no such thing as absolute safety because there will always be individuals attempting to get around your security features, but maintenance and updates are the best ways to keep your application secure. Each operating system has specific security features and regularly upgrades them to keep them secure.