Draw It or Lose It

CS 230 Project Software Design Template

Version 1.2

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	07/18/24	Sarah Tomlinson	<Updated and filled out Design Document>
1.1	8/4/24	Sarah Tomlinson	<Checked and updated Evaluation Section>
1.2	8/14/24	Sarah Tomlinson	<Updated Recommendations Section>

Executive Summary

The Gaming Room desires a web-based game that serves multiple platforms based on their current Android only app, “Draw It or Lose It.” The game consists of four rounds that last one minute each where a drawing is rendered at a steady rate and is fully complete at the 30-second mark. This is a team game where the team attempts to guess the puzzle (phrase, title, thing, etc.) within the time limit. If the current team does not answer within the time limit, another team will be given a 15-second time limit to solve the puzzle. The drawing will be rendered from a library of stock images to be drawn as clues for each puzzle.

Requirements

- A game will have the ability to have one or more teams involved.
- Each team will have multiple players assigned to it.
- Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
- Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

Design Constraints

- Teams must consist of multiple players
    - Teams will need to have a minimum number of players and should be distributed as evenly as possible to ensure fairness.
- Each team and game name must be unique to allow users to check if a name is in use when using a team name.
    - Team and game names must be saved so that they can be checked against each other to ensure that they do not match.
- The game must be run across multiple platforms
    - This requires that the game is made to be compatible across various web browsers and devices.
- Only one instance of the game can exist in memory at any given time.
    - Unique identifiers for each game, team, or player instance must be created
    - This means the design must also accommodate their being only one instance at a time for each aspect of the game (player, game, and team).

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

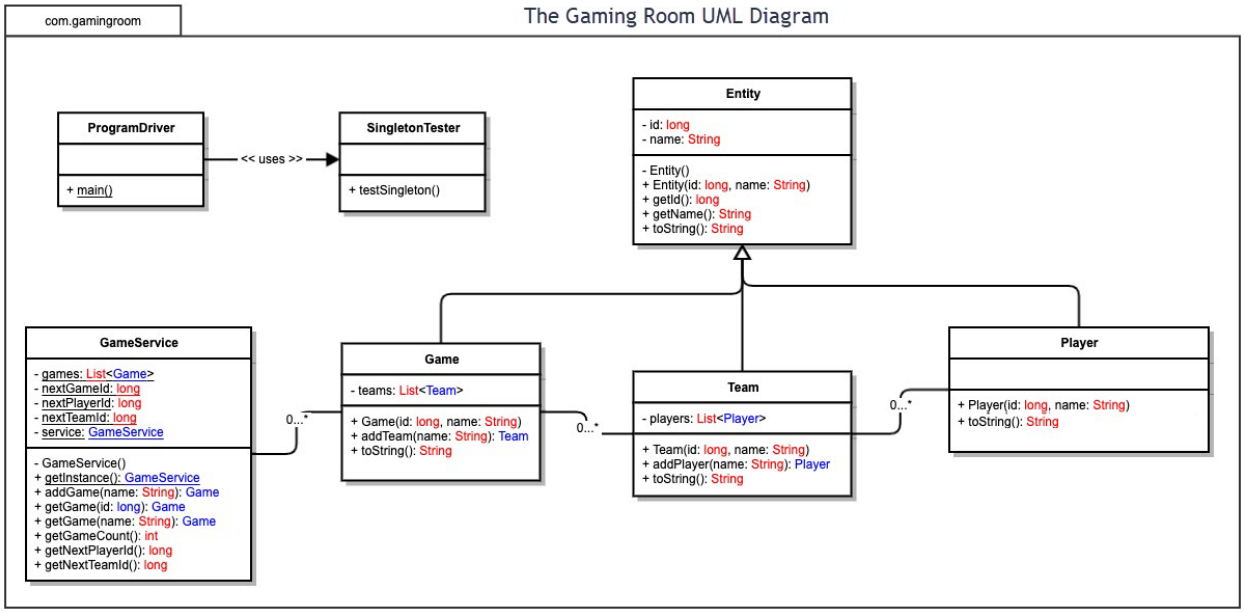
The UML Diagram below details the relationships between the various classes in the project.

The Entity class is the primary base (or parent) class; therefore, all other classes inherit information from the Entity class. The Entity class consists of 'id' and 'name' attributes, which are inherited from each of the subclasses. Thus, each entity in the system will have a unique identifier and name as determined in the Entity class. The Game, Team, and Player class each *extend* the Entity class, as shown by the arrow pointing from each of them towards Entity. As can be seen in the Game class, it includes multiple teams, which each includes multiple players.

Now looking at the GameService class which has a multiplicity (one to many) relationship with the Game class. That is, one GameService may include many games. In a way, it manages the Game instances and contains references to each of the Game objects.

The ProgramDriver class contains the main function, meaning this is where the program runs. This class has a directional relationship to the SingletonTester class represented by the arrow in the diagram. It also has the "<<uses>>" identifier, which means that the ProgramDriver class uses the SingletonTester class (indicated by the directional arrow).

The UML diagram showcases multiple object-oriented programming principles such as inheritance (between the Entity class and the Game, Team, and Player classes) and encapsulation (private attributes and methods marked by "-" that can be accessed by but not updated by other classes).



Evaluation

Development Requirements	Mac	Linux	Windows	Mobile Devices
Server Side	Mac is easy to use and set up servers. It contains a graphical user interface that allows for a variety of users to understand and utilize. It also comes with a steeper hardware cost than Linux and Windows. Mac OS X Server 10-client is $499. Mac OS X unlimited is $999.	Linux can allow for extreme levels of customization and flexibility. It uses command shells for server configuration and access but is inexpensive. Linux servers vary from $250 - $1300/yr depending on size and support needs.	Windows is a more expensive option but contains a wide range of software compatibilities. There is a large support network available to developers. Windows Servers range from $500 to $6200 per installation per year depending on user numbers, devices, and licensing model (Specialty servers vs. core-based)	Mobile Devices have the advantage of being portable but has a limited screen size. The hardware capabilities are variable depending on the specs of the hardware. It is harder to determine costs for mobile based servers due to a variety of factors, but estimates fall around $5 to $250 with services such as Heroku, Digital Ocean, etc.
Client Side	The interface is widely used and understood but is used less than other systems like Windows. Development requires a Mac with XCode.	Linux is difficult to use without prior technical experience. It does not use a graphical user interface like most client-side users are accustomed to. This system is used less than Mac and/or Windows.	Windows is widely used and understood but has a higher software cost than Linux or Mac. This is the most used system, and therefore would provide the most value across platforms.	There are limitations to design due to the smaller screen, but users are familiar with and able to use a wider variety of systems. Certain built-in features such as GPS and push notifications may aid development. Android uses Java and iOS uses SWIFT. iOS will need the same requirements as Mac, and Android can be refactored from Windows and Linux based servers.

Development Tools	HTML, CSS, Javascript and Node.js are some of the available languages. IDEs include VSCode and XCode. Also includes libraries to support front end development. XCode is $99/year per developer.	HTML, CSS, JavaScript, Ruby, PHP, and Python are some of the available languages. IDEs include VSCode, Atom, and Sublime Text. Also includes libraries to support front end development. Various IDEs can be used that are open-source/free. Eclipse, PyCharm are both free and cover Python, Java, and C/C++.	HTML, CSS, .NET Framework and C# are some of the available languages. IDEs include Visual Studio, Eclipse, and PyCharm. Also includes libraries to support front end development. Visual Studio for business use runs from $45 to $250 based on plan and features. Eclipse and PyCharm are free.	HTML, CSS, C++, Java and Javascript are some of the available languages. IDEs include Android Studio and XCode. Emulators can also be used. Also includes libraries to support front end development. XCode is $99/yr per developer.

Recommendations

1. **Operating Platform**: Windows will be the best fit for the operating platform for Draw It or Lose It. Windows is the most used operating platform and allows for the most flexibility in choice of language and IDE. Windows also can be utilized alongside the current build.

2. **Operating Systems Architectures**: Microsoft Windows uses a graphical user interface and provides a way to store files, watch videos, run various software and connect to the internet for browsing and playing games. It features a kernel mode and user mode. Kernel mode controls the memory and processing, while user mode handles applications. Windows uses a layered approach which also contains a layer to control hardware compatibilities (HAL). These features mean that Windows is a choice that allows for maximum versatility for the project. Windows has 32-bit and 64-bit variations that have similar architecture, but varying amounts of physical memory available.

3. **Storage Management**: Windows has a feature called Storage Sense. This basically just means that Windows allows you to manage the files on your hard drive, automatically freeing up space by getting rid of unnecessary files. Windows also features Storage Optimization and a Storage Management Console. The Storage Management Console provides a centralized location (graphical interface) for managing storage disks, volumes and files. Windows also uses NTFS (New Technology File System) to increase reliability and security for servers. This is the primary file system and employs security descriptors, encryption, disk quotas and rich metadata.

4. **Memory Management**: As mentioned before, the Storage Sense feature allows for easy management of memory space. This keeps everything needed for the game packaged together for ease of access and storage security. Windows has its own virtual address space for processes, where all threads have access to the visible address space of the process. The system provides a default heap for each process and private heaps help applications that make frequent allocations perform better. File Mapping is also a feature of Windows, which involves the association of file content with a piece of visible address space in the process. File Mapping allows processes to work effectively with large data files (like websites) but means that the entire file doesn't need to be mapped to memory.

5. **Distributed Systems and Networks**: A distributed system contains autonomous computer systems that are separated physically but are connected by a centralized computer network. For network-based interactions between multiple users, systems often use a shared database among the players that interact with each other over the network. For "Draw It or Lose It" The Gaming Room wishes to communicate between various platforms. My recommendations are to use a browser available on all platforms to run the application. Based on research, it looks like Google Chrome will be the best choice to use as it is available on every platform and is widely used and understood.

6. **Security**: Windows has its own security and firewall protection, but the user can also purchase and install their own preferred protection. It would be more ideal and recommended to use a different source for user information protection. One standard way is to secure with a username and password known only to the user as well as standard encryption techniques to keep user information safe as it is transmitted across the network. Windows also uses encryption for transferring data, increasing security for both the client and the server as data is transferred back and forth.