
<Alejandro Ochoa>

**CS 230 Project Software Design Template**

Version 1.0

**Table of Contents**



## Document Revision History

| Version | Date | Author | Comments |
|---|---|---|---|
| 1.0 | <01/28/2024> | <Alejandro Ochoa> | <Build web based game that can be used on multiple platforms. > |
| 1.1 | <02/11/2024> | <Alejandro Ochoa> | <Evaluate development needs and OS platform characteristics> |
| 1.2 | <2/25/2024> | <Alejandro Ochoa> | <Analyzed the characteristics of and techniques specific to various systems architectures> |

**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, a client of Creative Technology Solutions wants to develop a web-based version of their Android game, Draw It or Lose It. Key requirements are, that the game will have the ability to have one or more teams involved, Each team will have multiple players assigned to it, and 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.

**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.*

**Design Constraints**

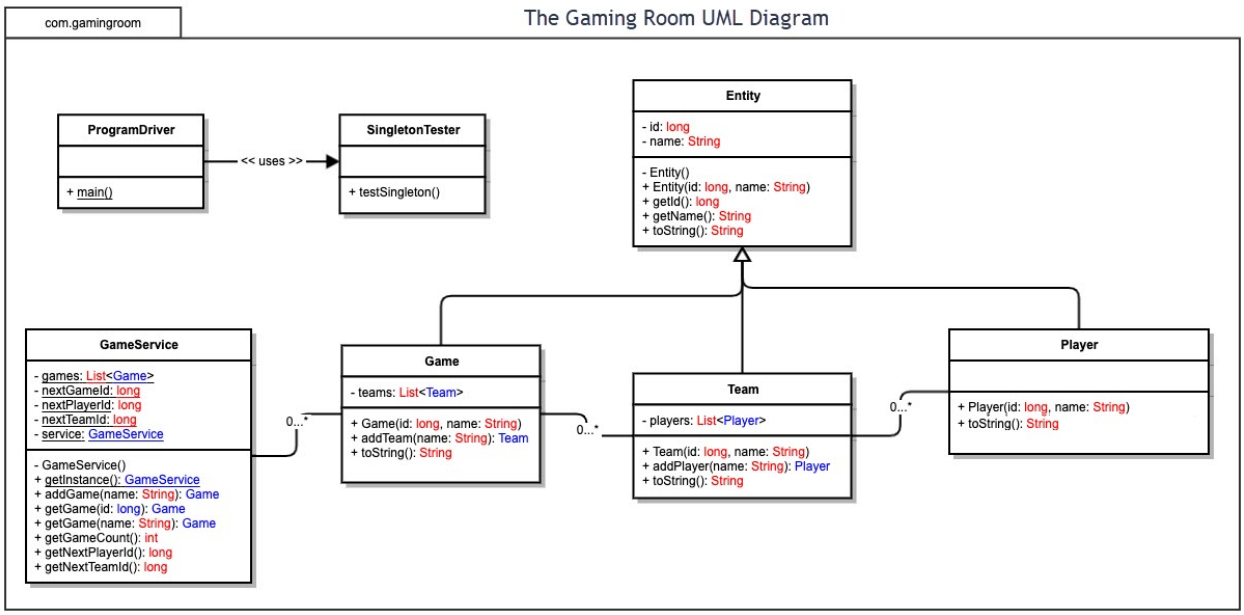
- *A game will have the ability to have one or more teams involved.*
- *Each team will have multiple players assigned to it.*
- *Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.*
- *Only one instance of the game can exist in memory at any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

**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 class diagram shows classes such as ProgramDriver, SingletonTester, Entity, GameService, Game, Team, and Player, along with their relationships. ProgramDriver uses SingletonTester to utilize its functionality. The entity class serves as a parent class, with Game, Team, and Player inheriting from it. GameService is paired with Game instances, while Game has Team instances, and each Team has multiple Player instances. The open arrow from Game, Team, and Player to Entity shows inheritance, where child classes inherit attributes. The diagram shows OOP  principles such as inheritance and aggregation.




## 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** | Easily accesible sever<br>Small pool of macOs users<br>Easy to use GUI<br><br>**Update2/11/2024:** macOS provides a Unix-based environment similar to Linux, which can benefit developers familiar with Unix systems. macOS Server has limited scalability compared to Linux and Windows Server. It may not be as well-suited for large-scale web hosting environments. macOS can be used for server-based deployment, it's less common than Linux and Windows. Tools like Apache and Nginx can be installed on macOS for web hosting. | Linux is free to use<br>Simple interface and not very mainstream<br><br>**Update2/11/2024:** Offers distributions like Ubuntu, CentOS, and Debian, each with its own set of tools and package managers. It's free to use. It's highly customizable, allowing for tailored server configurations. Linux servers are known for their reliability and performance. Linux does have a steep learning curve to it. Linux offers server-based deployment options through web servers like Apache, Nginx, and application deployment tools like Docker. | Easy to use<br>Most popular software and GUI<br>Expensive compared to others<br><br>**Update2/11/2024:** Windows Server is the most popular choice for hosting web applications. Windows Server offers seamless integration with other Microsoft products and services. It has robust support for .NET applications and tools like IIS for web hosting. Windows servers are more expensive in terms of licensing compared to other OS. Windows Server provides server-based deployment options through IIS and other web hosting platforms. | Hardware specs differ greatly from model to model<br>Portabilty<br><br>**Update2/11/2024:** Mobile devices are not typically used as server platforms. However, they can be utilized for client-side interactions through mobile apps. Mobile devices provide a platform for engaging with users directly through apps. Mobile devices lack infrastructure and resources required for server-side hosting of web applications at scale. They are more suited for client-side interactions. |

| Development Requirements | Mac | Linux | Windows | Mobile Devices |
|---|---|---|---|---|
| **Client Side** | Requires more learning on how to navigate<br>Expensive<br>Not as popular as windows<br><br>**Update2/11/2024:**<br>React Native is available for macOS.<br>Ensure communication between client-side applications and backend web applications by using communication protocols such as RESTful APIs. | Takes time and skill to navigate through linux<br>Very diverse<br><br>**Update2/11/2024:**<br>React Native is available for Linux.<br>Ensure communication between client-side applications and backend web applications by using communication protocols such as RESTful APIs. | Easiest to use<br>Expensive but MacOS is still more expesive<br><br>**Update2/11/2024:**<br>React Native is available for windows.<br>Ensure communication between client-side applications and backend web applications by using communication protocols such as RESTful APIs. | Lower hardware specs makes development and user usage diffcult when handling larger applications<br>Can use anywhere<br><br>**Update2/11/2024:**<br>Use cross-platform development like React Native to build mobile applications for both iOS and Android simultaneously. This reduces development time and cost by allowing code sharing between platforms. |

| Development Requirements | Mac | Linux | Windows | Mobile Devices |
|---|---|---|---|---|
| **Development Tools** | Programming languages Swift, C++, HTML, CSS<br>Uses Xcode apples offical IDE<br>Uses other tools such as, Homebrew, git, cmake | Programming languages used are Pyhton,C++<br>IDE used are, Visual studio code, Eclipse, Atom<br>Other tools use are Cmake, Git, G++ | Programming languages used are C, C++, Python, Javascript<br>IDE used are eclipse, Visusal Studio<br>Other tools used are Git, Windows SDK | Apples Mobile programming languages are Swift and Objective C<br>IDE is Xcode<br>Other tools they use are Fastlane, TestFlight<br>Android programming languages are Java and Kotlin<br>IDE is Android Studio<br>Other tools used are Firebase, Gradle, AndroidSDK |
| | **Update2/11/2024:** Swift is free to use. Other porgramming laguges and IDE have diffrent pricing. Multiple development teams specializing in different platforms may be required for consistency across each application. | **Update2/11/2024:** Visusal stuido code is free to use. Other porgramming laguges and IDE have diffrent pricing. Multiple development teams specializing in different platforms may be required for consistency across each application. | **Update2/11/2024:** Visusal studio is free to use. Other porgramming laguges and IDE have diffrent pricing. Multiple development teams specializing in different platforms may be required for consistency across each application. | **Update2/11/2024:** Xcode is free to use, as well as Android studio. Other porgramming laguges and IDE have diffrent pricing. |

**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 found that Windows OS will allow the game to expand to other computing environments. Windows has more software available and costs are more manageable. Windows has ease of use and access to a variety of software packages.

2. **Operating Systems Architectures**: Windows core components are integrated into the system. However, other aspects, such as device drivers and subsystems, can be implemented as separate modules. Windows offers flexibility and performance benefits for applications like Draw It or Lose It.

3. **Storage Management**: Windows has NTFS or New Technology File System as its default storage system. Additionally, cloud storage systems like Azure can be integrated for reliable storage.

4. **Memory Management**: Windows has virtual memory management, using a combination of RAM and disk space to create virtual memory. This allows apps like Draw It or Lose It to access more memory than what is physically available.

5. **Distributed Systems and Networks**: Distributed systems and networks offer easy communication between each other. But there are still common problems like connection issues, lagging computer performance and even routing problems. Draw It or Lose It can use RESTful APIs for real-time communication between clients on different operating systems. The network has failover mechanisms in place to ensure connectivity.

6. **Security**: Windows offers strong security features, it has role-based access control, encryption, and authentication mechanisms like Active Directory. Also If the Gaming Room wants more protection they can add a firewall and have a third-party team to conduct regular audits.