The Gaming Room

Draw it or Lose It

# CS 230 Project Software Design

Version 3.0

## Table of Contents

[Software Design Template 1](#_Toc21086455)

[Table of Contents 2](#_Toc21086456)

[Document Revision History 2](#_Toc21086457)

[Executive Summary 3](#_Toc21086458)

[Requirements 3](#_Toc21086459)

[Design Constraints 3](#_Toc21086459)

[Rationale 3](#_Toc21086460)

## Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 7/15/2023 | Chase Carter | Executive Summary |
| 2.0 | 7/26/2023 | Chase Carter | Requirements |
| 3.0 | 08/12/2023 | Chase Carter | Program Software Architecture |

## Executive Summary

*In summary, the purpose is to use the different approaches that have been explored to applying the software into different operating platforms. The integration of the game in a distributed environment has also been explored. We will now look at the actual application of the architecture for different operation platforms that are currently available. We will be looking to assign the users to a team with different identifications. There will be layering that will follow this process. This will make the game run smoothly. This process is to avoid issues with the user’s making teams.*

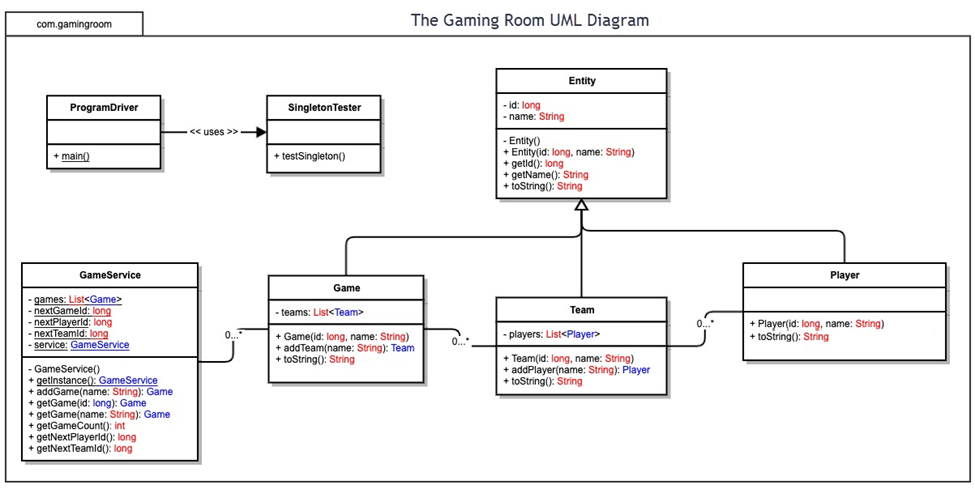
## Design Constraints

*The biggest design constraint will be the memory of the software of the game. The storage of the individual game and the users must be stored adequately to allow the continuation of the gameplay. The user’s games will be stored in a cloud-based memory. This will link to the appropriate server for storage. Each individual game must be removed from the memory upon completion. This is in order that the game will continue to run smoothly. The constraint of users will be limited to the number of players. The number of games being played at one time will be limited as well.*

## Rationale

*<Explain your rationale for each design constraint you’ve identified and how it relates to the requirements provided to you by the client.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirements | Windows | Linux | iOS | Mobile Devices |
| Server Side | By far Windows contains the most amount of software available. Out of all the other operating systems Windows has the most software available.  Some programs that can be applied are “react” and “flask”. | Very cost-effective compared to iOS even though the software is similar.  Some programs that can be applied for the memory are “Amazon Web Services” or “Digital Ocean”. | Programming is among the easier to use due to the flexible commands to configure the server. Some programs that can be applied are “WebSocket in HTML” and “Flask”. | Mobile applications are of the highest quality if the server is not located on the device. Some programs that can be applied are “react” and “flask”. |
| Client Side | Middle of the group concerning the cost-effective aspect. Entry or intermediate developer required. The aid for windows is extensive. Some software that can be used is “React”, “Flask”, and “VSCode”. | Very cost-effective. The level of difficulty for development is high due to the complexity of the operating system.  A program that can be applied is “react”. | Middle of the group concerning the cost-effective aspect. The operation system automatically updates. This is due to the OS using super plus. This allows the updates because there is no interaction with other OS.  A program that can be applied is “Flask”. | The design of the applications on the client side is very different due to the platform. There is cross-platforming. |
| Development Tools | Very extensive IDE to choose from. This goes for the development and the actual code. Most resources available will be Windows. | Like Windows, there is a wide variety of IDE to choose from. This is for web development and coding. | iOS is ahead with the available tools for development. There are not many choices due to the individual style of programming. | Cross-platforming is imperative concerning mobile devices. The frameworks must allow for screen compatibility when the application is developed. |



Architecture Recommendation:

1. Operation Platform:

By far Windows has the widest range of software available. This aids by having more support if problems arise in development. The price is not the cheapest but also is not the most expensive. The price is mid-range overall. The functionality of all the IDE is good. Some operate better than others. However, finding developers that can use Windows by far will be the easiest.

1. Operating Systems Architectures:

Graphical User Interface uses icons and menus to manage the overall function of the application. This is a built-in function of the Windows OS. Windows also allows the user of the application to message other users or other teams inside the application. There are many more web services that can be utilized inside of the application using GUI.

1. Storage Management:

The game system on the back end must allow the games to be erased upon completion to create room. There is a wide range of cloud-based memory tools. The price and capability vary depending on which one is chosen. There are some free options for cloud storage. Choosing the correct storage will depend on the supported operation system. However, I recommend Personal cloud storage or google play because this is directly affiliated with Windows OS. There is a free option, but it may not be sufficient in the amount of storage. I recommend paying for the 2TB, this will cost $10 per month. If more memory is needed, then more can be paid for even after the game has been released.

1. Memory Management:

There are many ways to go about managing the files that are created. A developer can create the necessary file for the game and compile and organize it manually. Or they can use a software development environment that’s designed for game development. I recommend UWP or Universal Windows Platform. This will aid in the actual management of the files that are created. This is mainly for Windows OS. There is a lot of support for memory management inside these development tools.

1. Distributed Systems and Networks:

For this application to grow into cross-platforming it must function with all operating systems. “Unity” is a good place to start development for cross-platforming. It is very cost-effective, and it is compatible with all other operating systems. A server that can support multiple games going on at the same time as well as allow the users to use the application without crashing is important. “ScalaCube” would be a good start for looking at game servers that are hosted. “ScalaCube” is a widely known and used host server and it provides plenty of power.

1. Security:

Windows has built-in security for the operating system. However, there is a concern when storing users' personal information, credit card information, and more. I would recommend a second line of defense apart from the building security. There is a wide range of software that is known and proven to be effective, I would start by looking at “Norton” and “McAfee”. This software will be looking for malware and viruses. I would also run a security diagnostic once the application is completed but before release. This can be done through a cyber security application, or a cyber security developer can be hired if the funds are available.

Citations:

stevewhims. “Windows Game Development Guide - UWP Applications.” Learn.microsoft.com, 16 Mar. 2023, learn.microsoft.com/en-us/windows/uwp/gaming/e2e. Accessed 12 Aug. 2023.

Cloudwards.net. “Cloud Storage Comparison 2019: Compare 50+ Cloud Providers.” Cloudwards, 12 July 2018, [www.cloudwards.net/comparison/](http://www.cloudwards.net/comparison/).

“9 Best Game Server Hosting Platforms for Everyone.” Geekflare, 12 Feb. 2021, geekflare.com/game-server-hosting/.