

Draw IT or Lose IT

# **CS 230 Project Software Design Template**

Version 1.2

## Table of Contents

1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/13/22 | Lawrence Arundel | Implementing entity class and functionality of gaming project. The revision includes a working demo. |
| 1.1 | 07/27/22 | Lawrence Arundel | Weighing the advantages and disadvantages of various project elements, such as the need for multiple types of client support, software development considerations (cost, time, expertise), and programming language and tool considerations. |
| 1.2 | 08/10/22 | Lawrence Arundel | Putting in place proper verifications, such as security, memory and storage management, and operating system architecture. |

**List of requirements and constraints to reflect upon throughout application development process:**

* 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. Accomplished by creating unique identifiers for each instance of a game, team, or player.

## [Executive Summary](#_sbfa50wo7nsh)

The software design problem indicated by the client reflects the need to integrate systems to run and execute a replication of the 1980s television game Win, Lose, or Draw. The game uses images stored within a library as each team solves the puzzle. The client wishes to have a game where one or more teams can be involved, each team having multiple players; game and team names being unique to allow users to check whether a name becomes utilized; and only one instance of the game can exist in memory. The business requirements for this project include developing a gaming app for multiple platforms using various software patterns in a distributed environment. The gaming app must be able to run on multiple platforms, including Linux, Mac, and Windows, as well as mobile platforms. The technical requirements include showcasing which software development tools become utilized to perform the business requirements, including what software will become utilized, conducting cost-benefit analysis, and other means to construct a well-developed game.

## [Design Constraints](#_2et92p0)

The design constraints for developing the game application in a web-based distributed environment include the distributed environment's having additional serialization and network latency overheads due to remote calls or RPCs (Silberschatz, 2003, s. 3.6.2). Other constraints include potentially being a more complex and more expensive environment in terms of total cost of ownership (Meier, 2007, p. 1). The implications of design constraints on application development include memory, battery life, ability to adapt to different screen sizes and orientations, security, and network bandwidth when considering mobile application development (Singh, 2022, p. 1).

## [System Architecture View](#_ilbxbyevv6b6)

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 become required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should become provided.

## [Domain Model](#_8h2ehzxfam4o)

The UML class diagram provided below shows us how classes are related to each other. Program Driver and Singleton Tester classes alignment by the associated solid black arrow. The <<uses>> specifies that Program Driver employs Singleton Tester to demonstrate that only one instance is applicable during execution. Game Service and Game have a binary association, meaning a relationship between two classes. The multiplicity for this relationship further denoted by 0...\*, meaning zero to many. Games and teams also have a binary association, meaning a relationship between two classes. The multiplicity for this relationship further denoted by 0...\*, meaning zero to many. A team and player also have a binary association, meaning a relationship between two classes. The multiplicity for this relationship further denoted by 0...\*, meaning zero to many. The classes of game, team, and player all show an inheritance relationship with the class entity. Inheritance means the mechanism wherein a new class derivation from an existing class (Techopedia, 2022, p. 1).

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | **Advantages:**  - Top-of-the-line hardware software is secure  - Apple’s elevated level of security protects data.  - Flexibility hosts Apache servers, which allow for basic web code to run smoothly.  **Disadvantages**:  - Closed platform (no manual updates)  -Cost (hardware and software updates)  (Long, 2022, p. 1) | **Advantages:**  - Low cost  - Easy server maintenance (Apache servers) to run basic web code (WordPress, forum software)  - Extremely reliable server uptime and security, and self-updating website and other software  **Disadvantages:**  - It requires extensive knowledge to operate efficiently and securely  (Long, 2022, p. 1) | **Advantages:**  - Windows can run multiple applications at the same time  - Host applications that Linux and Macs cannot (for example, ASP.NET, Microsoft Access, or MSSQL databases)  **Disadvantages:**  - Oftentimes, servers run slower and become more costly compared to the other options  - Limited to Windows applications  (Long, 2022, p. 1) | **Advantages:**  - Implementation of a mobile cloud  - Includes scalability affordability, hardware-free, improved speed,  - Disaster recovery (real-time backup)  **Disadvantages:**  -Implementation of a mobile cloud risks include security  - Data management  -Bandwidth limitations  (Long, 2019, p. 1) |
| **Client Side** | **Advantages:**  - N/A  **Disadvantages:**  -It is costly to obtain the hardware  -Very difficult to learn with minimal documentation  -Needs multiple screens to utilize optimal client-side experience (XCode)  (Kravis, 2021, p. 1) | **Advantages:**  - The cost is very negligible in comparison to Windows or Mac    - The time used to load web pages is quick because of JavaScript and other tools used within this software development environment (JavaScript)  (Lerner, 2016, p.1)  **Disadvantages:**  - Your expertise level to implement on Linux needs to be extensive | **Advantages:**  - It is ridiculously cheap compared to the Mac  - Supports multiple screens    - Is very intuitive  - Has few built-in features  **Disadvantages:**  - Windows Forms events are easy to detect (Visual Studio)  - Windows implements UNIX for communication in client-server systems.  (Kravis, 2021, p. 1); (Silberschatz, 2003, s. 3.1). | **Advantages:**  - The cost of creating and sustaining mobile environments is very inexpensive (monthly costs)  - The time used to implement client-side services is amazingly fast  - Efficient implementing JavaScript, CSS, and newly implemented QR codes  - The expertise to develop and use these services is very intuitive  (Fenlon, 2011, p. 1)  **Disadvantages:**  N/A |
| **Development Tools** | **Relevant programming languages for macOS include the following:**  - Python  - C++  - Java  (Kolakowski, 2020, p. 1).  **The IDEs used on Macs include the following:**  - XCode  - NetBeans  - Eclipse  **Game development software used for Macs includes the following:**  - Unreal Engine  - Google Cloud Platform  - Unity  - Adobe AIR  - Scratch  - Godot 2.0  - Gdevelop  - Torque2D  - Torque3D  (Computerscience.org, 2022, p. 1); (Hattersley, 2016, p. 1). | **Relevant programming languages for Linux include the following:**  - Python  - C  - C++  - Fortran  - Pascal  - COBOL  - Lisp  **The IDEs used on Linux include the following:**  - The Mono Project  - Lazarus  (Bolton, 2018, p. 1).  **Game development software used for Linux includes the following:**  - Unity  - Godot  (Computerscience.org, 2022, p. 1); (Smith, 2022, p. 1). | **Relevant programming languages for Windows include the following:**  - C++  - C#  - C  - XML  - HTML  - CSS  - PHP  - JavaScript  **The IDEs used on Windows include the following:**  - Visual Studio  - NetBeans  - Eclipse  (Veracode, 2022, p. 1).  **Game development software used for Windows includes the following**:  - GameMaker  - Studio 2  - Unity  - Unreal Engine 4  (Computerscience.org, 2022, p. 1); (The Art Institute, 2017, p. 1) | **Relevant programming languages for Mobile Devices include the following:**  - Objective-C  - Swift  - Python  - C/C++  - Java  - JavaScript  (Fireart Studio, 2022, p. 1).  **The IDEs used on mobile devices include the following:**  - Android Studio  - XCode  - Visual Studio  - IntelliJ  - Xamarin (AppVerticles, 2020, p. 1).  **Game development software used for mobile devices includes the following:**  - Unity  - Unreal Engine  - Solar2D,  - Fusion 2. 5  (Computerscience.org, 2022, p. 1); (Kriebernegg, 2022, p. 1) |

## Recommendations

1. **Operating Platform**: I would recommend using Windows to develop this style of game for The Gaming Room to implement.
2. **Operating Systems Architectures**: The details of Windows include its flexibility within architectures, including web, client, and development tools. Windows provides flexibility and has a steady learning curve, meaning most developers would be comfortable working on this operating system. Windows also provides clear efficiency with this style of game and robust features, including the use of multiple applications run simultaneously and can host servers, which Linux and Mac cannot. Knowing the operating system, we can further choose the style of architecture. Implementing Microservices means a service-oriented architecture well suited for this application. It gives the developer more control over the development process and overall functionality of the application than a service-oriented architecture (Arsov, 2021, p. 1).
3. **Storage Management**: I would recommend implementing Azure Functions as a serverless cloud computing storage system for being cheap and effective, setting up different environments at ease, and options with automatic scalability (Bashir, 2019, p. 1).
4. **Memory Management**: Windows has its own virtual address space for each 32-bit process, allowing up to 4 gigabytes of memory which is viewable. Each process has an 8-terabyte address space on 64-bit Windows. All threads have access to the visible address space of the process. Threads, on the other hand, do not have access to the memory of another process, which protects one process from becoming damaged by another. Other qualities of memory management include heap functions and file mapping. Heap functions can help applications that make frequent allocations from the heap perform better. File mapping allows the process of input and outgoing sequences (I/O) of the file map to work effectively with large data files, such as websites, without requiring the entire file mapped to memory (Clintra, 2022, p. 1).
5. **Distributed Systems and Networks**: A distributed system is a collection of processors that do not share memory or a clock. (Silberschatz, 2003, s. 16). Windows utilizes distributed systems and networks called interprocess communications (IPC). These IPCs include clipboards, COM, Data Copy, DDE, File Mapping, Mailslots, Pipes, RPC, and Windows Sockets. Utilizing RPC enables applications to call functions remotely. Windows also utilizes data sharing between applications. Storing information via Azure creates a more robust experience and secures data in case of outages or other circumstances that would inhibit data transfer capabilities (Russinovich, 2020, p. 1).
6. **Security**: Windows security pertaining to Windows includes two-factor authentication capabilities to keep a connection very secure between the user and the server. Windows/Microsoft also includes certain coding languages for developers to add these commands to their code that will identify threats and stop them before they get too far (DEP) or data execution prevention (Malik, 2015, p. 1). The PREfast tool allows programmers to quickly find bugs that would otherwise become overlooked by compilers, programs that change one coding language to another. The Azure cloud service allows you to build an encrypted IPSec (Internet Protocol Security) tunnel from the user side. Azure provides malware protection and multi-factor authentication services that you can use as a standalone server during on-premises integrations (Prov international, 2021, p. 1).

References:

AppVerticles, A. V. (2020, December 23). *Best ides for mobile app development: Our top 5 picks*. Appverticals. Retrieved July 4, 2022, from <https://www.appverticals.com/blog/best-integrated-development-environments/>

Arsov, K. (2021, May 18). *Microservices vs. SOA - is there any difference at all?* Medium. Retrieved July 4, 2022, from https://medium.com/microtica/microservices-vs-soa-is-there-any-difference-at-all-2a1e3b66e1be

Bashir, F. (2019, July 13). *What is serverless architecture? what are its pros and cons?* freeCodeCamp.org. Retrieved July 4, 2022, from https://www.freecodecamp.org/news/what-is-serverless-architecture-what-are-its-pros-and-cons/

Bolton, D. (2018, May 25). *Best programming languages for linux DEVS*. Dice Insights. Retrieved July 4, 2022, from https://insights.dice.com/2015/03/09/best-programming-languages-linux-devs/

Clintra, C. (2022, February 1). *Windows memory management*. GeeksforGeeks. Retrieved July 4, 2022, from https://www.geeksforgeeks.org/windows-memory-managment/

Computerscience.org staff, C. (2022, June 23). *Guide to Computer Programming & Coding languages*. Code a New Career | ComputerScience.org. Retrieved July 4, 2022, from https://www.computerscience.org/resources/computer-programming-languages/#python

Fenlon, W. (2011, June 27). *How mobile web pages work*. HowStuffWorks. Retrieved July 4, 2022, from <https://computer.howstuffworks.com/mobile-web-page.htm>

Fireart Studio, F. S. (2022, January 17). *Top most popular programming languages for mobile app development*. Fireart Studio. Retrieved July 4, 2022, from https://fireart.studio/blog/top-most-popular-programming-languages-for-mobile-app-development/

Hattersley, L. (2016, July 6). *The 19 best game development software solutions for mac 2022: Reviews of the most popular tools & systems*. The 19 Best Game Development Software Solutions for Mac 2022 | Reviews of the Most Popular Tools & Systems. Retrieved July 4, 2022, from https://www.capterra.com/game-development-software/s/mac/

Kolakowski, N. (2020, December 2). *7 programming languages popular at Apple that could land you a job*. Dice Insights. Retrieved July 4, 2022, from <https://insights.dice.com/2020/12/01/7-programming-languages-popular-at-apple-that-could-land-you-a-job/>

Kriebernegg, T. (2022, March 17). *The best 22 Mobile Game Engines and Development Platforms*. App Radar. Retrieved July 4, 2022, from https://appradar.com/blog/mobile-game-engines-development-platforms

Kravis, S. (2021, August 30). *Software development on windows and macos*. TurboFuture. Retrieved July 3, 2022, from https://turbofuture.com/computers/Software-Development-on-Windows-and-MacOS

Lerner, R. M., & Python, R. L. teaches. (2016, June 30). Client-Side Performance. Retrieved July 4, 2022, from https://www.linuxjournal.com/content/client-side-performance

Long, R. (2022, February 5). *Windows hosting vs Linus hosting which is better for you?* Web Design Leads & Resources. Retrieved July 3, 2022, from https://www.omegaweb.com/windows-v-s-linux-v-s-mac-web-hosting/

Long, S. (2019, September 27). *Mobile cloud hosting: Pros and cons to consider*. 7T, Inc. - SevenTablets. Retrieved July 3, 2022, from <https://7t.co/blog/mobile-cloud-hosting-pros-and-cons-to-consider/>

Malik, Muhammad. (2015, May 25). *Ways Developers Keep Computer Games secure*. Information Security Buzz. Retrieved July 4, 2022, from https://informationsecuritybuzz.com/articles/ways-developers-keep-computer-games-secure/

Meier, J. D. (2007, December 1). *Web application performance design guidelines - deployment*. Web Application Performance Design Guidelines - Deployment - Guidance Share. Retrieved July 3, 2022, from <http://www.guidanceshare.com/wiki/Web_Application_Performance_Design_Guidelines_-_Deployment#Distributed_Architecture>

Prov international, P. V. I. (2021, July 12). *4 Microsoft Azure Security features to give you peace of mind*. ProV International - The Best Technology Consulting Firm. Retrieved July 4, 2022, from https://www.provintl.com/blog/4-microsoft-azure-security-features-to-give-you-peace-of-mind

Russinovich, M. R. (2020, August 17). *Advancing the outage experience-automation, communication, and transparency: Blog E atualizações do azure: Microsoft Azure*. Blog e atualizações do Azure | Microsoft Azure. Retrieved July 4, 2022, from https://azure.microsoft.com/pt-br/blog/advancing-the-outage-experience-automation-communication-and-transparency/

Silberschatz, A., Galvin, P. B., & Gagne, G. (2003). *Operating system concepts*. Hoboken, NJ: Wiley.

Singh, A. (2022, May 3). *Essential aspects to consider while designing mobile apps*. GlobalLogic. Retrieved July 3, 2022, from https://www.globallogic.com/insights/blogs/essential-aspects-to-consider-while-designing-mobile-apps/#:~:text=The%20main%20constraints%20are%20memory,%2C%20security%2C%20and%20network%20bandwidth

Smith, O. (2022, January 24). *Linux game development on ubuntu: Godot and unity*. Ubuntu. Retrieved July 4, 2022, from https://ubuntu.com/blog/get-started-with-linux-game-development-on-ubuntu

Techopedia, T. (2022, June 3). *What is inheritance in java? - definition from Techopedia*. Techopedia.com. Retrieved July 3, 2022, from <https://www.techopedia.com/definition/3226/inheritance-java#:~:text=Inheritance%20is%20a%20mechanism%20wherein,derived%20is%20called%20a%20superclass>

The Art Institutes, T. A. I. (2017, July 10). *3 game design software tools you can use to make your own games*. 3 Game Design Software Tools You Can Use to Make Your Own Games. Retrieved July 4, 2022, from https://www.artinstitutes.edu/about/blog/aig-3-game-design-software-tools-you-can-use-to-make-your-own-games

VeraCode, V. C. (2022, January 1). *What is IDE or Integrated Development Environments?* Veracode. Retrieved July 4, 2022, from https://www.veracode.com/security/integrated-development-environment

YouTube. (2017). *Uml Class Diagram Tutorial*. *YouTube*. Retrieved July 4, 2022, from https://www.youtube.com/watch?v=UI6lqHOVHic&ab\_channel=LucidSoftware.