

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 | 03/19/23 | Marc Aradillas | Initial release |

**Executive Summary**

The Gaming Room is a company that wants to develop a web-based distributed game application called Draw It or Lose It. The game is intended to be played on different computing environments and mobile devices. The purpose of this software design document is to provide a clear understanding of the requirements, design constraints, and recommendations for developing the game application. The document includes an explanation of the UML class diagram, which demonstrates how the classes relate to each other and how object-oriented programming principles are used to fulfill the software requirements efficiently. The document also evaluates different operating platforms, operating system architectures, storage management, memory management, distributed systems and networks, and security. Based on the evaluation, the document recommends the appropriate operating platform, operating system architectures, storage management system, memory management techniques, and security measures to be used for the Draw It or Lose It game application.

**Requirements**

The client's business and technical requirements for the Draw It or Lose It game application are as follows:

* The game application should be web-based and distributed, which means it can be played on different computing environments and mobile devices.
* The game application should be able to handle multiple users simultaneously.
* The game application should be able to store user data securely.
* The game application should be able to handle user inputs efficiently and respond to them in real-time.
* The game application should provide an enjoyable user experience.

**Design Constraints**

The design constraints for developing the Draw It or Lose It game application in a web-based distributed environment are as follows:

* The game application should be able to handle different computing environments and mobile devices with different hardware and software configurations.
* The game application should be able to handle different network conditions, such as latency, bandwidth, and connectivity.
* The game application should be able to handle different security threats, such as unauthorized access, data breaches, and denial-of-service attacks.

**System Architecture View**

There is nothing required in this section for this project

**Domain Model**

The UML class diagram for the Draw It or Lose It game application shows seven classes: ProgramDriver, SingletonTester, Entity, GameService, Game, Team, and Player. The ProgramDriver class and the SingletonTester class are not related to the game logic, but they are used to start the application and test the Singleton pattern, respectively. The Entity class is a superclass for the Game, Team, and Player classes. The GameService class is responsible for managing the game, which includes creating and updating games, teams, and players, and handling user inputs. The Game class represents a game session, which includes a collection of teams and players. The Team class represents a team, which includes a collection of players. The Player class represents a player, which has a name, a score, and a team. The UML class diagram demonstrates several object-oriented programming principles, such as inheritance, encapsulation, and association. The Entity class is a superclass for the Game, Team, and Player classes, which demonstrates inheritance. The GameService class encapsulates the game logic and manages the game objects, which demonstrates encapsulation. The association between the GameService class and the Game, Team, and Player classes demonstrates association.

**Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement** | **Linux** | **Mac** | **Windows** | **Mobile Devices** |
| Server Side | Linux is often used as a server operating system due to its stability, security, and scalability. It has excellent support for web and database servers, making it an excellent choice for the server side of a web-based game application. | Mac OS is built on Unix and is known for its stability and security. However, it is not commonly used as a server operating system, and it may not be the best choice for the server side of a web-based game application. | Windows Server is a popular choice for hosting web-based applications due to its excellent support for Microsoft technologies, including .NET and SQL Server. It is known for its ease of use and extensive support for third-party software. | Mobile devices are not suitable for server-side development, as they do not have the necessary processing power and resources to handle the demands of a web-based game application. |
| Client Side | Linux is not as widely used on the desktop as Windows or Mac OS, but it is still a viable platform for desktop game applications. It has good support for the OpenGL graphics library, making it a desirable choice for games that require 3D graphics. | Mac OS is a popular platform for desktop game development due to its high-quality hardware and excellent support for OpenGL graphics. It is a viable choice for developing games that require advanced graphics capabilities. | Windows is the most widely used desktop operating system and has excellent support for DirectX graphics library, making it a good choice for game development. It also has a large user base, which can be advantageous for the distribution of the game application. | Mobile devices are increasingly popular platforms for game development, with both iOS and Android offering robust development tools and a large user base. However, developing games for mobile devices can be challenging due to the limited processing power and screen size of the devices. |
| Development Tools | Linux has a wide range of development tools, including popular integrated development environments (IDEs) such as Eclipse and NetBeans. It also has excellent support for open-source programming languages such as Python and Ruby. | Mac OS has excellent development tools, including Xcode, which is the official IDE for developing Apple applications. It also has dedicated support for programming languages such as Swift and Objective-C. | Windows has a wide range of development tools, including Visual Studio, which is one of the most popular IDEs for game development. It also has excellent support for programming languages such as C# and Visual Basic. | Both iOS and Android offer robust development tools for mobile game development, including Xcode and Android Studio, respectively. They also have their own programming languages, Swift and Kotlin, respectively, which are specifically designed for developing mobile applications. However, developing games for mobile devices can be challenging due to the limited processing power and screen size of the devices. |

**Recommendations**

1. Operating Platform: Based on the evaluation, we recommend that The Gaming Room use a web-based platform for Draw It or Lose It. This will allow the game to be accessible on any device with a web browser, including desktops, laptops, tablets, and smartphones. A web-based platform also provides the advantage of easy maintenance and updates as they can be implemented on the server-side, without requiring user-side updates.
2. Operating Systems Architectures: The recommended web-based platform can be designed with a three-tier architecture, consisting of a presentation tier, an application tier, and a database tier. The presentation tier will be responsible for rendering the game interface, while the application tier will handle game logic and user authentication. The database tier will store game data, such as user profiles and high scores.
3. Storage Management: For storage management, we recommend using a cloud-based solution, such as Amazon S3 or Google Cloud Storage. These services provide scalable, reliable, and secure storage for data, with options for backups and disaster recovery.
4. Memory Management: The recommended web-based platform uses memory management techniques such as garbage collection, which automatically deallocates memory that is no longer needed by the application. This ensures that the application uses memory efficiently and avoids memory leaks, which can cause the application to crash.
5. Distributed Systems and Networks: To achieve communication between various platforms, we recommend using a RESTful API, which can be implemented on the server side. This API can be accessed by different devices and platforms, allowing them to communicate with the server and share game data. In case of network outages, the API should handle retries and fallbacks to ensure continuity of the game experience.
6. Security: To protect user information, we recommend implementing industry-standard security practices such as secure user authentication, data encryption, and access control. The recommended web-based platform can use SSL/TLS encryption to secure data in transit and store user passwords in hashed form to prevent unauthorized access. Additionally, the platform can implement measures to prevent common web application attacks such as SQL injection and cross-site scripting.