

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)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 | 03/20/2022 | Drew Pepin | Filling out the executive summary, design constraints, system architecture view, and domain model. Also evaluating design requirements for different software. Ill also be recommending design ideas to The Gaming Room. |

**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](#_sbfa50wo7nsh)

The Gaming Room has hired us to develop a web-based game that serves multiple platforms. The game is called Draw It or Lose It, and is currently only available as an android app. The game is played in 4 rounds lasting a minute each. Drawings are rendered at a steady rate, fully completing in 30 seconds. If the team does not guess correctly, the opposing teams have an additional 15 seconds to guess.

## [Design Constraints](#_2et92p0)

* Needs multiple teams
* Teams are made of 2 or more people
* Team names must be unique
* Only 1 game can exist in memory at a time
* Be able to run on multiple platforms

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

The Game, Team, and Player classes form a relationship by inheriting information through Entity. Each of these classes use name and id as references within their particular classes. Team and Player are “has a” classes. Game has a team, with GameService having a Game. GameService references Games, Games reference Teams, and Teams reference Players. So you add a player, you then add that player to a team, then add that team to a game.

**"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** | Mac servers are an advantage to use if you’re within a network of other Mac users. It offers a flexible terminal, and is a popular choice for web hosting. Also has a user friendly interface. | Linux is more difficult to navigate and learn compared to Windows/Mac. Fairly simply command operation. Known for being steady/powerful/reliable. | There are far more available software options on Windows than Mac/Linux. Has a user friendly interface. Windows is gaining popularity in web hosting. | Finding a mobile provider for your backend code is extremely difficult. Security can be an issue when doing this as well. |
| **Client Side** | Average time and expertise required to use Mac. The cost can be more than Windows, but different price points available. Skills needed to handle OS are simple. | Minimum cost compared to Mac/Windows. Requires a lot of time and expertise to master Linux, but considered amongst programmers as an elite operating system. | Pricier than Linux, slightly cheaper than Mac. Easy to learn, not a lot of expertise or time required. | More difficult to implement that other options like Mac/Windows/Linux. Easier to access. Can checks servers for updates from anywhere. Requires a lot of skill/time. |
| **Development Tools** | Dev Tools include using VisualStudio, eclipse, notepad++, GitHub. Languages include HTML/CSS/JS, and using libraries to help you run languages like Python/React/PHP/Java. IDE’s for Mac can be JS, Python, PHP, Ruby. | Dev Tools include visual studio code, eclipse, notepad++, GitHub. IDE’s for Linux include JS, Python, PHP, Ruby. Can also run HTML/CSS. Libraries to support front end languages like React/Python/PHP. | Dev Tools are Visual Studio code, eclipse, notepad++, GitHub. C++ is the most popular language with Microsoft systems. IDE’s for languages utilized for Windows OS are Python, C++, HTML. Can also run HTML/CSS/JS, and has libraries to support Python/PHP/React/Java. | Tools you can use include VSCode, Eclipse, notepad++, Github. Can use languages such as HTML/CSS/JS, and libraries for Python/PHP/React/Java. |

## Recommendations

1. **Operating Platform**: When it comes to choosing the appropriate operating platform for Draw it or Lose it, I believe finding an operating platform that is flexible and easy to use is the best approach. With that said, I am recommending Windows for Draw It or Lose It. Windows has the most amount of software options available as requires minimum expertise. Windows also offers an easy to use and user friendly interface. Windows is also in the mid-range for cost compared to other options.
2. **Operating Systems Architectures**: Window’s offers the user the ability to work with several different languages, allowing the user to pick whatever they’re most comfortable with. Window’s GUI also offers many Window’s based applications to help with your development. You can also easily manage memory storage on Windows.
3. **Storage Management**: Window’s offers useful features when it comes to storage management. When it comes to mobile applications, Window’s allows for the application to be stored on your phones main memory after installation. This can greatly save storage space, as well as result in faster processing. Window’s also has to ability to handle cloud storage, another useful tool to save space on the hardware of the device.
4. **Memory Management**: With Draw It or Lose It, the game requires 200 photos to be used on memory during gameplay. In order to manage this, Windows offers management options with virtual or physical address space, with up to 4 gigabytes of memory allowing applications to run efficiently when using memory.
5. **Distributed Systems and Networks**: When it comes to distributing your software so that can it be used on multiple platforms / devices, Window’s offers different IDEs you can explore that will help you run Draw It or Lose It on any device or platform. You can find options that allow you create your application, then export to IOS, Android, etc… Having cross play will open up many options for Draw It or Lose it.
6. **Security**: Window’s allows their clients to protect their information and data going in / out of their application. Window’s offers security features such as Anti-spyware, anti-virus, VPN and more. All of these security checks happen in real time while running the application, preventing malicious threats to penetrate your application and software.