

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)4

[**Recommendations**](#_m8aleynsvzvc)7

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/18/2022 | Thomas Fiske | This revision is to add information needed by Game Room to create Draw It or Lose It. This includes the Executive Summary, Design Constraints, System Architecture View, Domain Model, Evaluation, and Recommendations. |
| 2.0 | 4/3/2022 | Thomas Fiske | This revision was to add the evaluation table after learning new information about different operating systems and how they function. |
| 3.0 | 4/15/2022 | Thomas Fiske | This revision was to add the recommendations section of the document. |

## [Executive Summary](#_sbfa50wo7nsh)

The game Draw It or Lose It is being created by the company The Gaming Room. The game will render images from a large library of stock drawings to use as clues for players in the game to guess what the puzzle is. The puzzle can consist of a phrase, a title, or a thing. The game will have four rounds with each round lasting one minute each. The drawings will render overtime and eventually be fully rendered by the 30 second mark. The opposing team has a chance to guess within 15 seconds if the original team does not guess correctly within the time limit.

This game will be developed as a web-based application. It is currently only available on Android devices. In order to achieve this. The application will have to run on multiple different operating systems including Mac, Linux, Windows, and Mobile Devices (Apple, Android). For this to work, the application will have to be made using a coding language that is easily transferable between these platforms.

## [Design Constraints](#_2et92p0)

* The application must run on multiple platforms
* The application must be able to create unique identifiers for each game, team, and player.
* The application will need to be able to render an image slowly and then fully at a set time
* Only one game instance can exist at any one time
* There will be multiple teams and players inside of those teams.
* The application will have to run fast enough to react to players within time limits of the game.

## [Domain Model](#_8h2ehzxfam4o)

This UML class diagram shows that GameService is has a relation with Game. We see that it says “0…\*” This means that GameService is associated with 0 or more objects in game, and that every object in Game is associated with one object in GameService. Game also has this relation with Team, and Team has this relation with Player. These classes all inherit from each other in order to create the game, the teams inside of that game, and the players inside of those teams. The Game, Team, and Player classes are all connected to the class Entity, this is an inheritance relationship. This means that Entity is the parent class of Game, Team, And Player and all of these classes inherit from the Entity Class.

**"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** | | Powerful server operating system that is supported by Apple. Every user with a Mac account has the ability to host websites. With the support of Apple, it is possible for Mac based web applications can host millions of players at a time. There is licensing cost associated with Mac. | Open sourced and a cost-effective operating system to host websites on. No licensing fees to host on Linux. There is no major cooperating that is backing Linux, but it is able to host web applications. It does not have the power to host web applications on the scale as Apple, Google, or Windows. | Windows provides support for multiple operating environments. Windows is also supported by Microsoft. There are certain technologies that can only be used when hosting with windows such as ASP, MSSQL and more. There are licensing fees associated with Windows. Like Apple and Google IOS’s, Windows has the power to host millions of players at a time. | Mobile OS are not as strong as the other three competitors.  IOS is supported by Apple.  Android is supported by Google. There are licensing fees for both Android and IOS operating systems. The servers have the power to host millions of people at a time. |
| **Client Side** | | Mac has a lot of products that only work with their systems. For example, if one team member has a mac, then everyone should have a mac to effectively use the devices together. There would have to be quality assurances tests on any Apple product that was being planned to run on another OS. | Linux is affordable and downloadable on a lot of different computers. It is open sourced as well. All applications developed with Linux should be tested on different platforms to assure portability. | Windows is an easy to grasp OS that has portability and connectivity that the rest of the OS do not have. It can be run on different types of computers as well. The portability and ease of use of windows makes development easier and can cut down on development time. | Limited on the number of IDE’s and tools available to create and maintain a web-based application using a mobile OS.  There would need to be tests to make sure that the application runs smoothly on computers compared to mobile devices. There can be display issues when trying to use a mobile hosted application on a computer. The web application might be limited by the processing power of a mobile device as well. |
| **Development Tools** | | * Swift * Objective-C * C++ * C * Assembly Language * Atom IDE * Microsoft Visual Studio IDE * AppCode IDE * CLion IDE | * C * Assembly Language * Eclipse IDE * Atom IDE | * Python * C++ * C * JavaScript * Java * CSS * HTML5 * Eclipse IDE * Microsoft Visual Studio IDE * Atom IDE * ASP * MSSQL | * Swift * Java * Koltin * Microsoft Visual Studio IDE * Android Studio IDE * XCode IDE |
| **Development Tools (cont.)** | All Apple products have a licensing fee to use for application development. A company that makes an application that will be hosted by Mac will have a lot of the tools to host an IOS web application as well because Apple owns both of these operating systems. This means that in the future if a company wanted to switch to IOS hosting instead of Mac hosting or vice versa, they would have the machines and tools to do so because they use the same machines and tools. | All tools on Linux are free because it is opened source. Linux has a lot of community support so there is ample opportunity for developers to search for help (if needed) in Linux communities. | Most of these tools can be used for free, but things like ASP and MSSQL require paying a licensing fee. There are many different tools that can be used by developers using Microsoft web hosting. The portability and ease of use of windows will cut down on training and development time. There are also many different languages that can be used with windows. This gives a lot of freedom and choice about how the application will be created and developed. | Applications for mobile devices have different operating systems that they can be built on. This means that they can be used with a few different languages depending on the company. This gives a lot of flexibility to a company when deciding how to make their application. A company with the tools to host on IOS, would have a lot of the tools needed to host on Mac, so there is a lot of portability between the two. A lot of Android development tools are free to use online. There is also an in-depth guide by Android on how to build applications that could help developers in a pinch or train new employees. |

**Recommendations**

1. **Operating Platform**: Windows is the recommendation for Draw It or Lose it. This is because Windows is easy to use, and has a lot of portability and connectivity. It is also supported by Microsoft which means that it will continue to be supported and have functionality in the future.
2. **Operating Systems Architectures**: The architecture of Windows two layers, a user mode and a kernel mode. The user mode allows applications to be run on the operating system while the kernel mode has access to the systems memory and devices that are connected to the operating system.
3. **Storage Management**: Windows has applications that allow the user of the operating system to view and manage their storage. The user can delete files, add files, change files etc. It also shows how much memory each type of file takes up as well as how much memory each file takes up.
4. **Memory Management**: Windows uses applications that send virtual memory addresses to Ram and uses virtual memory to help run programs faster. Windows servers are robust and able to handle and provide the memory needed to run this application and save and process user data.
5. **Distributed Systems and Networks**: Windows has connectivity and portability and can be used with a lot of devices. Windows devices can work well with devices with other operating systems. Windows is also supported by Microsoft so it is regularly updated.
6. **Security**: Windows is support by Microsoft and this means that the security is also supported by Microsoft. This makes Windows a very secure operating system that has systems in place to protect user data, as well as solve any security issues should they arise.