

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 | 09/16/2021 | Rico Applewhite | First Version |
| 2.0 | 10/01/2021 | Rico Applewhite | Filled in Evaluation |
| 3.0 | 10/17/2021 | Rico Applewhite | Filled in Recommendation |

**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 wants to develop a web-based game that is base on their current game (Draw It or Lose it). In this game, teams tries to solve puzzles as an images slowly renders. The game is currently only available on an Android app and the company want this web-based game to serve multiple platforms. The Gaming Room wants the game to involve one or more teams with multiple players in each. The Software has to check and make sure each Game, Team and Player have unique name and make sure there is only on instance of the game existing in the memory at any given time

## [Design Constraints](#_2et92p0)

Multiple Players: The game will involve multiple players playing with and against each other. This will require the software to be able to run seamless for each and every player who will likely be miles away. I suggest that games, team and player limit are implemented so not to overload the software ,for example if thousands of player tried to join one game. The software will also need to be able to keep track of all the players.

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

## [Domain Model](#_8h2ehzxfam4o)

In the UML class diagram below, inheritance plays a big part of the code. Player, Team and Game class all inherits from the Entity base class, which stores the important variables that many of the class will use. In this case Id and Name. The Player, Team and Game classes will use these variable to assign new names and ids as well as search and retrieve existing one. This will help fulfill the software requirement of being able to keep track of the players.

**"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)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac can support web server software, Licensing cost is higher than Linux, but not quiet as much as Windows | Linux does have a way to deploy web hosting. Licensing cost is the lowest compared to the other operating system. One weakness is choosing a good distributor. Some service last for years while other may last for a few months. | Windows also support web server software. There are more options, but Licensing on windows cost significantly more than Linux and Mac. | Mobile supports some web server software. |
| **Client Side** | Development for Mac cost more than Linux. Not as technical to work with as Linux but significantly more than windows. May require a bit time to learn | Linux is very cost efficient, however is very technical to those without prior knowledge of the system. May require a significant amount of time to learn | For windows, software development is cost efficient. Windows is one of the most common used os, so most will have prior knowledge and time should not be an issue. | Development for mobile can be costly. More technical knowledge will be needed and it will be harder to integrate it with the other three operating system  However,With the Andriod App already made, development cost, time and expertise shouldn’t be to much compared to building a new game from scratch. |
| **Development Tools** | Common Programming Languages for mac includes C\C+ and Java and IDEs includes Visual Studios and Eclipse | Programming languages that can be uses includes C \C+, Python and Java  IDEs also includes Visual Studios and Eclipse | Windows Programming languages includes C/C+ Python and Java  IDEs also includes Visual Studios and Eclipse | Unity # is a programming language that is most popular in developing mobile games |

**\*Extra Notes:** Even though each operation system use the same Development tools I believe multiple development teams are needed to work on the each operation system, but work closely with each other to makes they are compatible with one another.

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: I Recommend Windows as the operating platform for The Gaming Room to expand “Draw it or Lose it” to other computing environments. Its the most common used and has access to more IDEs then other Operating platform. Through working from this platform, it will be more flexible creating a multi platform software
2. **Operating Systems Architectures**: Two architecture of windows are 32 bit and 64 bit. 32 bit are compatible with older systems, but are less secure. While 64 is faster and has better performance
3. **Storage Management**: Windows uses disk managements to manage storage. Some features includes compression data to take up less storage and the ability to extend data if more need to me added to it. Storage management is a important feature for storing all the graphics that will be use for the game as well as all user data.
4. **Memory Management**: Memory Management for Windows includes the compression features. Some other feature includes Prefetch, which loads important information before it is needed. This will be a useful feature in the “Draw it or Lose it” software for management many aspects of the game including, the slowly revealing of the photo in each round.
5. **Distributed Systems and Networks**: Distributed System and Networks will play a big part in the players being able to play with each other. Game Performance, Security and functionality will be based on these system and Network.
6. **Security**: To protect user information for the software, windows provides various of security features. These features includes “Windows Defender Exploit Guard””Microsoft Bitlocker” enhances system’s security even further “Windows Defender Credential Guard” which can protect data even if the organization's network is compromise