

Braxton Morrow

# **CS 230 Project 2 Software Design**

Version 3.0

## 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 | 07/15/21 | Braxton Morrow | Updated information to give project recommendations and overview. |
| 2.0 | 08/02/21 | Braxton Morrow | Went over information, since I filled out too much of the project last time, went over info to double check accuracy and added more to the evaluation. |
| 3.0 | 8/12/20121 | Braxton Morrow | Reviewed information a final time, added some changes to the server architecture recommendation and info |

## [Executive Summary](#_sbfa50wo7nsh)

The gaming room currently owns and operates an android app called, “Draw It or Lose It” and wants to develop the game rebranded onto multiple platforms the purpose of this game is that multiple teams consisting of several people, play four rounds at one minute each. Each round consists of a picture being pulled from a preset library of drawings. The drawings are rendered steadily and are finished after 30 seconds. If the first team doesn’t guess correctly within the time limit the next time has a chance to steal a point, but only gets 15 seconds to do so.

## [Design Constraints](#_2et92p0)

* Game needs to run on multiple servers and platforms
* Must be coded on something like Java to allow to transfer to other platforms
* Minimum internet speed of 3 Mbs
* Only on instance of the game will be allowed at any time

The first design constraint when using a web-based distributed environment is that Creative Technology Solutions is going to have to have multiple machines to host the servers for this game they wish to create, another is that this program is going to need to be coded in something like Java that can be translated across multiple platforms when desired for later use. Consumers of the game are going to be required to have a minimum internet speed of 3 Mbs.

## [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)

The ProgramDriver class includes main and is the framework to run the program. This is in the singleton format as to make sure only one instance of the game exists in a specific time. Entity creates relationships between Game, Team and Player class. Game service is inherited by Game. So essentially GameService has Game, Game is inherited from Entity, and has a relationship with Team, and Team has Player, all again under Entity.

****

## 

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| Server Side | Flexible terminal commands to configure the server, access or make changes  Can also use a Mac OS X server for $20 a month | More cost friendly than Mac or windows. Linux is open source meaning no operating systems fees.  Best security, and reliability for server structure  Many people are not as familiar with Linux between the other two | More software available compared to other OS. Linux is a bit simple with what you can do, and not all programs run on Mac OS, however Windows is most popular OS, so more programs.  Servers will most likely require frequent reboots, and not the best security | Definitely the most difficult to do as it would be better if it could be tracked at a single place. |
| **Client Side** | Moderate expertise and time required. The least used OS, typically you only want to use Mac when you’re working on an art medium. Cost a bit more than windows. | Minimum expertise and time required. However not everyone is familiar with Linux. Minimum Cost | Minimum expertise and time required. Most people are familiar with windows as it is the most popular OS. Cost a bit less than Mac. | Allows for clients or developers to update whenever desired, again harder to implement over other operating systems. |
| **Development Tools** | Swift is probably the most popular option for Mac, however many people still use notepad ++. However Macs can run all languages. | With Linux you could use Visual Studio, eclipse, or notepad ++. Again any of the more popular languages are usable with Linux, such as Javacript, HTML, CSS, C++, Java, Python etc. | My personal favorite for coding, easier use than Linux, and can run any language.  Personal favorite IDE’s are Visual Studio for compiled languages and Visual Studio Code for non-complied like node.js | Android and swift are going to be common use for mobile as android and MacOS are the most common phones. |

Recommendations

1. **Operating Platform**: I would recommend The Gaming Room starts on windows devices as it has more software available along with minimum expertise required to start coding the project. I would also argue that windows has the most IDE’s to work with.
2. **Operating Systems Architectures**: Windows provides services used by all windows-based applications to show a Graphical User Interface, while accessing system resources.
3. **Storage Management**: Windows 10 comes with a feature called storage sense. This allows you to better manage the files on your hard drive, showing how much space the item takes up when it was installed etc. You can also choose where the storage is saved to, making things easier to find if you want to access them quickly. Setting these in a stack would also be helpful as it would allow you to better move and remove pictures from the data set as needed.
4. **Memory Management**: For this project, you will need to create a pre-set library on a server somewhere to store the preset drawings for the renderings. The memory allocations for windows allow you to set the library somewhere easily accessible for later use outside of the default picture folder.
5. **Distributed Systems and Networks**: Personally, the IDE you pick is a preference, however I believe Visual Studio to be the best IDE for most jobs, you can code in most languages quite simply or download the packages to code in the rest. To fix problems like outages and connectivity you will need to ensure that you have enough server real estate to cover your needs and your players needs, as well as maintain some semblance of backup power to ensure your game never goes offline.
6. **Security**: Windows comes with built in security software, though to secure user data it would be recommended to use an outside security company. However, the server data should be safe enough with windows since it has its own system scan for malicious software should the need arise. Ideally this is why most people use Linux for their servers, however I am confident in my recommendation. Windows OS is the superior option here.