

# **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 | 05/22/2021 | Isley Baggett | Updates to all portions of document. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room, a game development company, wants to bring their popular android game “Draw It or Lose It” to a web-based platform. They want this game to serve multiple platforms, however, there current staff does not know how to set up the environment, which is why they have bought us, Creative Technology Solutions, into the loop.

## [Design Constraints](#_2et92p0)

Tools to build a game that can work on multiple OS will be a constraint in this case. The game is strictly android, and they want this to be able to work on a wide range of devices to include mobile devices. What hardware we have available could become a restraint however, the company has informed us this information will follow our proposal.

## [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 this UML diagram GameService, Game, Team, and Player are all associated and feed into each other. Player will help build Team, which in turn will feed into Game, and finally will end in GameService. All four of these classes will also inherit everything from their parent element Entity except for GameService which is not related to the Parent class Entity except by association. Under Game, Team, and Player you are creating Objects which are use through the get() method allowing for those instances to be used effectively. They are required because this will allow you to build your teams to play against each other fulfilling one of the requirements set forth by The Gaming Room.

****

## [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** | <Evaluate Mac for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Linux for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Windows for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> |
| **Client Side** | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mac.> | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Linux.> | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Windows.> | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mobile Devices.> |
| **Development Tools** | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mac.> | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Linux.> | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Windows.> | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.> |

## 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 would recommend that Linux be the operating system that the GameRoom goes with. I choose this because its open-source background will make it cost effective for the developers and has plenty of material outside of the documentation to help developers. While it may require some expertise from a Linux programmer its light weight will help with the speed of the application.
2. **Operating Systems Architectures**: There are two shells offered by the Linux operating system graphical shells and command-line shells. The graphical shell offers GUI interfaces for the user; however, this does come at a slower speed then that of a normal command shell. The portability of Linux allows for many different types of hardware and environments. It also allows for multiple applications to run simultaneously and for multiple users to have access to the same applications.
3. **Storage Management**: For storage Linux has a Hierarchical file system. Meaning that it will give structure to user files and system files. Logical Volume Manager or LVM is a Raid-like system available for Linux which allows you to create “drop boxes” for a lack of better terms for data and can be changed over time to allow for more storage and adapt to the needs of Draw it or lose it.
4. **Memory Management**: Memory management is one downside of management because it will require a programmer with the proper knowledge to integrate as it has its own set of “jargon” as the documentation puts it. Zones is just one way you could manage the memory in draw it or lose it by coding the current picture on ZONE\_NORMAL it would take up kernel space for processing and then the rest of the unused images could be in the background in a ZONE designated as unimportant at the time. There is also page cache which will allow data to be accessed more quickly after it is read for the initial time.
5. **Distributed Systems and Networks**: Cross platform Virtualization would allow for a developer to create Draw it or Lose it on one operating system, in this case Linux. This would stop the need for porting which if done poorly can ruin a user’s experience. Instead, the operating instructions would be “translated” into other operating systems/devices with no need to port. When it come to network have a back up provider if your current provider can not handle your user base is a must as an outage or connectivity issues would be frustrating. Another recommendation is having an option for when power goes out. This could either be a generator for the building or UPS that can allow your program to be run even with power interruptions.
6. **Security**: Linux is considered secure form the get go but to ensure your users never have issues. One recommendation is to code your application so that when a user logs on it gives rights to a temporary user so that in never links directly back to the user’s device. This would allow for a gap in between the user’s information and the information that potential hackers would see.