

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[CS 230 Project Software Design Template](#bookmark)1

[Table of Contents](#bookmark1)2

[Document Revision History](#bookmark2)2

[Executive Summary](#bookmark3)3

[Design Constraints](#bookmark4)3

[System Architecture View](#bookmark5)3

[Domain Model](#bookmark6)3

[Evaluation](#bookmark7)3

[Recommendations](#bookmark8)5

## [D](#bookmark9)[ocument Revision History](#bookmark9)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | <01/22/2022> | <Tatsiana Labkovich> | <Brief description of changes in this revision> |

**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.

## [E](#bookmark10)[xecutive Summary](#bookmark10)

The client wants to develop a current web-based game to run on multiple platforms. The game is called Draw it or Loose it, and currently available on Android. The client has some specific software requirements to implement when designing the game. The game should have more than one team involved, each team will have multiple players, all the names should be unique, with only one instance of the game.

To achieve this, our team must produce distinctive identifiers for each instance of a game, team, or player.

[Design Constraints](#bookmark11)

* The game must perform effectively on different platforms.
* The game will have the ability to have more than one team
* Multiple players on each team
* Only one instance of the game can exist at any given time

## [System Architecture View](#bookmark12)

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.

## [D](#bookmark13)[omain Model](#bookmark13)

The UML diagram includes set of classes that related to “The Gaming Room” : Entity, ProgramDriver, SingletonTester, GameService,Game, Team and Player.

The main object-oriented principle that we can observe here is inheritance. “Entity” class is the parent class with three classes inheriting properties from it. Classes “Team” and “Player” have “has a” relationship, and “Game”, “Team” and “Player” have “is a ” relationship. “Entity ” class controls access to public methods and protects data, therefore encapsulation also is seen on our diagram.

**"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.**

## [E](#bookmark14)[valuation](#bookmark14)

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** | -Able to run MacOs , Windows and Linux apps next to each other  -quality hardware  -less preferred for web hosting services (use Apple hardware only, more expensive than other operating system) | Cost friendlier than Mac, provide the best security, but not a lot of applications available.  Deployment is supported by .NET servers  Is the most popular operating system for web servers. since its open source, it is cheaper and easier to use. | The best platform to develop the game, user friendly, have some security flaws.  Windows offers The Web Deployment tool that offers deployment and management of Web Servers | The biggest issue is security, it is better if server is static, but on the plus side newest and portable platform |
| **Client Side** | * Cost comparable with Windows * Average expertise * Average time | -Minimal cost  -Maximum expertise  -Maximum time | -Cost similar with Mac  -Average expertise  -Minimal time | * The most cost effective * Average expertise * Maximum time |
| **Development Tools** | Mac supports all languages, some of them are  HTML  CSS  Javascript  Python  PHP.  Supports Visual Studio, NotePAd++, Eclipse  Pricing will be higher using MacOs | Same as Mac;  C is default language, Preferred tool would be Eclipse. | Same as Mac.  Windows is budget friendly. | HTML, CSS,Javascript.  Android is the great platform for creating applications. |

## 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**: The best operating system to develop the specific game would be Windows. It offers wide variety of IDE, minimal cost and expertise.
2. **Operating Systems Architectures**: Windows is the very popular operating system. Windows offering higher performance and wide variety of games.
3. **Storage Management**: Windows offers cloud server storage.Also, Windows comparable with database management system, which will be the most effective.
4. **Memory Management**:The game will use a lot of images and videos, in order to store them in one place we will need to create a database. So data management system that mentioned earlier will become handy.
5. **Distributed Systems and Networks**: Distributed systems and networks often have some issues. Security, compatibility , synchronization are some of them. By applying extra hardware, resources these issues can be controlled.
6. **Security**: Windows provide build in security protection software. Data encryption is widely used to provide proper security for clients data.