

"Draw it or lose It"

**CS 230 Project Software Design Template**

Version 1.0

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0  1.2 | 02/20/2021  04/4/2021 | Albert Castle | Used different Operating platforms as examples of how the program could move forward and be changed for future updates. |

**Executive Summary**

The client wants a game that has team guessing what the other is drawing or being sketched. The client has asked that this game offers it users a milt team game with custom player IDs. The client only wants one instance of the game, to do this we will need to have identifiers for each instance of a game, team or player.

**Design Constraints**

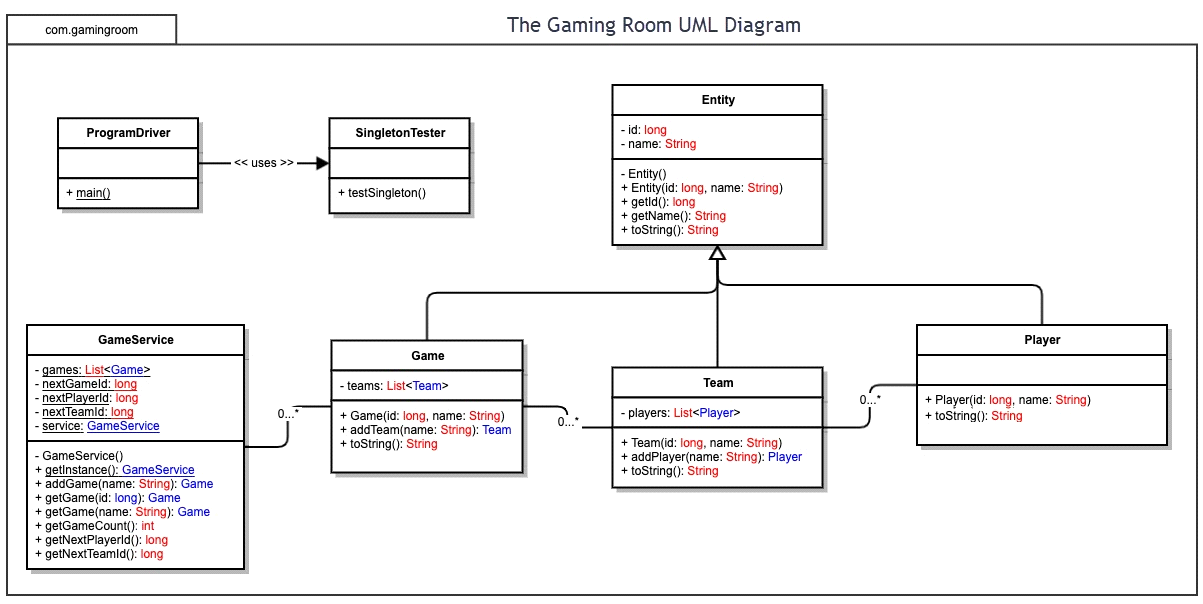
We need to be able to have one or more teams involved in the game. Each team is also going to have multiple people on it because one person will be drawing the sketch while the other tries to guess what it is. There can only be one instance of the game going on at one time. This application must be able to run on multiple platforms. To do this, we would need to use different software that is able to run code for Apple applications and any other platforms that could be used. We could possibly use multiple programing languages to do this, but we will have to ensure that our software is secure. Using multiple languages can bring some security concerns.

**System Architecture View**

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. Communication and storage aspects is also necessary to understand the overall architecture and should be provided.

**Domain Model**

This UML class diagram below includes some classes. Three of the classes in the diagram (game, team and player) show the principal of inheritance (you can create new classes that build upon existing classes) because they are also listed in the Entity class. The game, team and player classes have a "is a" relation to the entity class, this means that in Java, this relation is used for code reusability, this will help expand and make changes to the game.



**Evaluation**

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 must 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, like windows, is a closed platform which means that they can release updates. * There servers require higher end software, so this price is generally higher | * Linux is a free software that anyone can change, (open source) * Mac and Linux both have command shell that is used for server configuration. | * Along the same lines as Mac, Windows is a closed platform which would require similar, high end software that will cost more. | * Mobile specification is going to vary because a lot of phone run on Android or IOS software which would create a challenge to get them working reliably across both platforms. |
| **Client Side** | * Some expansive knowledge of this software should be required. * Mac and windows are owned by companies (Microsoft, Apple) and this results in a high cost of licenses | * If the choice is going to be based on price, the Linux will be the best option. It is an open-source operating system that you can get for free. | * Like Mac, Windows is owned by a larger company (Microsoft) and this will include a high cost of licenses as mentioned in the Mac slide. | * Provides flexibility to developers to update anywhere there is a connection. * Doing this will be slightly more difficult than Mac, or Windows. |
| **Development Tools** | * Swift is a software from Apple that is used for game development, XCode is apples IDE used to develop software for macOS. * its license is Freeware with open-source components. | * “Unity” is the most popular gaming engine in the world it works on Mac, Windows and Linux. you can create 2D and 3D multiplatform games. * We will be able to use the IDE "Visual Studio Code." | * There are many programming languages that could be used for Windows, one being Java. * As stated in the UML diagram description, Java uses a "is a " relationship that makes the code reusable | * One Programing language that can be used for Mobile devices is JavaScript. This language is good for games. * Since JavaScript is an important component for the internet it is already a very stable programing language for making games such as this one. |

**Recommendations**

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

* **Operating Platform**: If we want to expand on "Draw it or lose it," the operating platform that should be chosen is Windows. Windows has a good variety of applications that we could use to expand the qualities of the games by using different IDE’s for the software
* **Operating Systems Architectures**: Windows can be purchased online and downloaded onto whatever PC someone is working on. Windows has a lot of software packages available to give the programmers flexibility with the software they want to use.
* **Storage Management**: A feature that is not limited to Windows but an important tool, is cloud storage. This will be useful because the people making this software will never have to worry about running out of space, and it gives the flexibility to get more if need be. We could also use cloud-based storage. The games data will be called from the cloud when the user is trying to load them game up, the data will be sent to the user form the cloud so they can play. This will also be good for our users because no data will be on their computer, that to will also be stored in the cloud. The memory is something that will be more on the developer of the application as they will need to ensure that the game runs fast. We can do this by memory mapping a file, this will allow a part of the virtual address space to be associated with a file, this could lead so significant performance increases.
* **Memory Management**: Windows has options when it comes to memory management, you can buy physical space, or you can also buy space for memory online and download it (depending on the specs of your computer). This will ensure that you have enough GB of memory to run your applications without it crashing all the time. The game engine can provide many different libraries which have already been built so that will make it easier for us to work without having to hard code everything into the system. To obtain better memory space we can use a process called dynamic loading. With this process, a routine is not loaded until it is called. All routines are kept on a disk in a relocatable load format, it will not be loaded unless it is called. The advantage of this would be that the portion of the “Draw it or lose it game” that is being used and loaded will be smaller than the total program size. This is especially useful because it will not require any special support from the operating system and will help with any errors that might pop up.
* **Distributed Systems and Networks**: Distributed Systems are used in electronic banking systems, multiplayer online video games. Distributed systems are known to be able to handle more requests than a standard system. This will be good for the game because as stated by the client, this application needs to be able to run on multiple platforms. Most distributed systems will consist of a socket which is defined as an end point for communication. A socket is identified by an IP address, and then communicated between a server.
* **Security**: Windows 10 provides users with built in protection with no extra cost included. These settings can be changed by the user and can be app specific changes. Big changes that are made need to be authorized by the administrator. Windows also has new built-in features such as Microsoft Direct access so people can work remotely. There is an authentication process that you must go through when you connect to specific networks.  When setting up the program. Passwords are going to be key in getting users safely on the site, when the user is going to identify themselves, if the user supplied password matches with the password stored in the system, it will think that the owner of the password is using the system, but as we know people’s passwords can get stolen so that will need to be figured out.

**References**

F. Adelstein, S. Gupta, et al. “A Brief Introduction to Distributed Systems.” Computing, Springer Vienna, 1 Jan. 1970, link.springer.com/article/10.1007/s00607-016-0508-7.

Gewirtz, David. “Windows, Mac, or Linux? We Compare the Pros and Cons of These Computing Platforms.” ZDNet, ZDNet, 8 Apr. 2019, [www.zdnet.com/article/windows-mac-or-linux-we-compare-the-pros-and-cons-of-these-computing-platforms/](http://www.zdnet.com/article/windows-mac-or-linux-we-compare-the-pros-and-cons-of-these-computing-platforms/).

m, Michael. “Native or Web-Based? Selecting the Right Approach for Your Mobile App.” UX Magazine, 2014, uxmag.com/articles/native-or-web-based-selecting-the-right-approach-for-your-mobile-app.

Point, Java. “Inheritance in Java - Java point.” [Www.javatpoint.com](http://Www.javatpoint.com), [www.javatpoint.com/inheritance-in-java#:~:text=Inheritance%20in%20Java%20is%20a,behaviors%20of%20a%20parent%20object.&amp;text=The%20idea%20behind%20inheritance%20in,fields%20of%20the%20parent%20class](http://www.javatpoint.com/inheritance-in-java#:~:text=Inheritance%20in%20Java%20is%20a,behaviors%20of%20a%20parent%20object.&amp;text=The%20idea%20behind%20inheritance%20in,fields%20of%20the%20parent%20class).

Chardon, Lindsay. “Best Programming Languages for Game Development.” Gumede Academy, 13 Mar. 2021, gamedevacademy.org/best-game-development-languages/.

“Operating System Concepts, 8th Edition.” O'Reilly Online Learning, Wiley, learning.oreilly.com/library/view/Operating+System+Concepts, +8th+Edition/9780470128725/silb\_9780470128725\_oeb\_c03\_r1.html#h6.

