Creative Technology Solutions (CTS)
CS 230 Project Software Design Template
Version 1.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	07/15/23	Tayla Ashbrook	Producing a web-based game for multiple platforms called “Draw It or Lose It”

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

Producing a web-based game for multiple platforms. “Draw It or Lose It” is played by rendering images from a large library of stock drawings as clues so that the players in each team will work together to guess the Puzzle before the time runs out. One game consists of four rounds, one minute long. The Drawings are rendered at the 30 second mark. If the team doesn’t get the right answer before the time expires each member of the competing team has 15 seconds to guess to gain the points.

Requirements

- Played as one or more teams
- Each team will consist of multiple players
- Unique game and team names to identify if the names are already being used
- Create unique identifiers for each instance of a game team or player to save in memory (Only one instance of the game can exist in memory at any given time)

Design Constraints

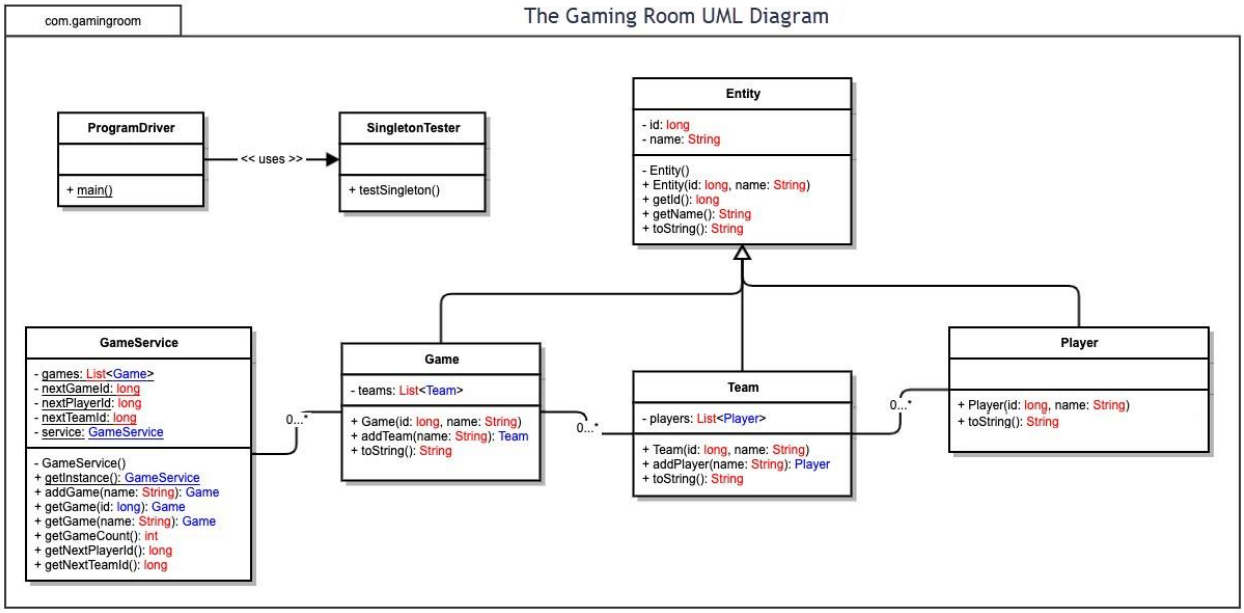
We need to produce a code that will run on all devices and platforms. So, if the game is already available on android phones, then we need to move forward on how to make it compatible with other mobile devices and machines such as Windows, Linux, and Apple.

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. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

Domain Model

The game, team, and player classes are connected or create a relationship with the Entity class as an inheritance. Each class shares a common reference that carries through each class creating the attributes of each induvial game. The GameService references the game, the game references the team, and the team references the player. The classes grab or inherit the information from the Entity class. So, when creating the game, the input attributes or accessors(getters) are stored in the Entity class then separates the attributes into their own class that builds the program to store and sets the information into the GameService. Each attribute must be created uniquely so that they can be identified correctly.



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 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	ADVANTAGES -Easy creations of features -Flexible commands to configure the server -Easy to make changes -Very popular characteristics -Upgradeable -Multiple options for different web requirements DISADVANTAGES -Not preferred for web hosting -Limited hardware options -Some productivity software are unavailable	ADVANTAGES -Stability -Reliability -Easy to use -Unix based -Linux servers' internet has been running for years without failure -Secured -Most preferred -Cost-friendly -Security flaws are caught before they become an issue DISADVANTAGES -Not commonly supported on applications so the web hosting is required other needs	ADVANTAGES -More software are available -Windows hosting which supports hypertext markup language files. -Uses characteristics that are dominant to other platforms -less loading time -Windows is becoming more common and popular -High resource requirements -High comfortability DISADVANTAGES -Poor tech support -Vulnerable to virus, scams, and hacking	ADVANTAGES -Portable -Cost effective -characteristics ae popular DISADVANTAGES -Harder to code -Harder to find a host that will support -Lengthy process -Poor security -Specifications are better on other devices

Client Side	-Mac users are forced to buy the MAC system built by Apple -Costs are similar to windows -Moderate expertise -Moderate time required -Answer may vary based on client's experience and preferences	-One of the best operating systems -Linux compatible computers are fairly common -No financial costs but are a few internal options based on user so there are minimum costs -Most of the software are open-sourced -Maximum expertise -Maximum time required	-Great platform for websites and window apps -Visual studio -Minimum expertise -Costs similar to mac -Minimum time required -Easier to use compared to Linux	-Mobile devices include cell phones and tablets -Flexibility -Though using a mobile device can be quite similar to actual computes and machines they can be different and a little more difficult to get all the information that are provided on the actual website and software -Less screen -Mobile platform UI controls
Development Tools	-Swift is a popular option -Mac can run all languages for example: HTML, JavaScript, Python, PHP, etc. -Objective-C is commonly used -Notepad++	-Supports most programming languages -Visual studio -Eclipse -Notepad++	-Many of the core applications are produced using C++ -Visual studio -Eclipse -Notepad++ -compatible with most programming languages	-Swift framework -Still uses most of the programing languages

SERVER SIDE:

When dealing with multiple platforms it is good to find all the details of each platform to get a general idea of what you will be looking at and working with including the problems you may run into so that it can be prevented, and a solution can be found to give you the results that are intended. Things to look at on the Server Side before starting your project may be:

- Characteristics: Key characteristics of each operating platform such as Performance, stability, security, and difficulty.
- Advantages: Advantages that each operating platform holds that can help identify what each platform has to offer.

- Weaknesses: The limitations of each platform.
- Server- based Deployment: note whether each platform offers a server-based deployment method and how they will handle the required scale for multiple players.
- Potential Licensing Costs: Initial cost, ongoing licensing fees, and any other cost information needed to know for each server operating system of each platform.

CLIENT SIDE:

The application must be delivered as a modern, responsive HTML interface running inside the web browser for desktop clients (Linux, Mac, Windows), as well as on mobile platforms. Each will be capable of communicating with the back-end web application running on the server. It is important for the client to notice all the considerations dealing with each platform including:

- Costs: The client will have to look at the Developing and testing for each platform, although there might be similarities between the platforms there may be additional resource and efforts that are required from the developing team that can impact the costs. Setting a budget for the developing team and requesting frameworks such as Cross-platform development can keep the costs a reduced amount.
- Time: The time will be based on Complexity of each platform and requirments that the development team need to complete the tasks. So, the timeline needs room in case of problems between multiple platforms.
- Expertise: Each operating platform requires different skills and complexity that makes them different that the developers will need experience on to help create and ensure that is works correctly.

DEVELOPING TOOLS:

Each of the operating platforms uses their own unique programming languages and tools or software's to code. When designing a code on a preferred operating platform it would be good to know the knowledge of the coding/ programing languages, the IDEs and Tools as well as the experiences with these requirements that would impact the cost, the time, and problem solving. In addition to the Tools that the platforms use there may be costs for some unique features based on the licensing that may be needed.

Mac:

Server Side:

- Platform Characteristics: The flexible terminal commands are useful for configuration of the server and to make changes. It is a very popular web hosting. It is user-friendly and reliable. The macOS Server offers Time Machine backup, file sharing, and network services.
- Platform Advantages: There are multiple web hosting requirements, with really good security. Using a user-friendly interface that is easy to navigate and is compatible with other Apple products. It stays up to date with upgrades
- Platform Weaknesses: Although it is a very popular web hosting it is also the least preferred for web hosting services and is a smaller user base compared to Windows
- Server-based Deployment: macOS Server
- Potential Licensing Costs: Moderate

Client Side:

- Cost: Similar to Windows
- Time: Moderate
- Expertise: Moderate

Developing Tools:

- Programming Languages: JavaScript, Python, Ruby, and HTML/CSS/JavaScript
- IDEs and Tools:
    - Development: Visual Studio and Xcode
    - Server-Side: PyCharm or WebStorm
- Impact: Very popular with developers that are familiar with Mac platforms and macOS environments. If the developer has a great understanding and experience with macOS tools it can reduce the time and impact the costs because they will know certain shortcuts that can bring what is expected with the least amount of code.
- Licensing Costs: Some mac development tools are free or have affordable licensing options. For example: Xcode and Visual Studio may have licensing costs for certain features.

Linux:

Server Side:

- Platform Characteristics: Preferable for stability, security, and scalability
- Platform Advantages: Security flaws are caught before they become an issue and just like the mac it is a preferred choice for web hosting services but great for server environments
- Platform Weaknesses: It is a more difficult platform therefore it could be a learning curve for developers and is more difficult to find applications to support the required needs. In this platform some of the software may have limited support.
- Server-Based Deployment: Commonly used for servers
- Potential Licensing Costs: Lower to Free licensing costs based on the application that is used, which with Linux most are open sourced and free to use.

Client Side:

- Cost: Minimum
- Time: Maximum
- Expertise: Maximum

Developing Tools:

- Programming Languages: JavaScript, Python, PHP, and HTML/CSS/JavaScript and more that are like mac and windows
- IDEs and Tools:
    - Development: Visual Studio
    - Server-Side: PyCharm and WebStorm
- Impact: Most of the Linux tools are open sources with the proper amount of experience using these tools impacts the developing process and decreases time needed to complete the project.
- Licensing Cost: Since most of the development tools are open-sourced and free to use this reduces licensing costs. But using Visual Studio there may be some Licensing Cost depending on the features that are needed.

Windows:

Server Side:

- Platform Characteristics: Windows is another popular choice for web hosting using a Windows-based environment. This platform is dominant compared to the other platforms.
- Platform Advantages: Windows is easily manageable and has great support through Microsoft technologies. The platform is easy to navigate through and has a high comfortability.
- Platform Weaknesses: security is lower than Linux and mac and may have higher security risks and concerns.
- Server-Based Deployment: Windows Server
- Potential Licensing Costs: Moderate to High

Client Side:

- Cost: Similar to mac
- Time: Minimum
- Expertise: Minimum

Developing Tools:

- Programming Languages: JavaScript, Python, C++, HTML/CSS/JavaScript, and .NET
- IDEs and Tools:
    - Development: Visual Studio, Eclipse, and .NET
    - Sever-Side: PyCharm and WebStorm
- Impact: Expertise in .NET programming language that is a popular choice when using Visual Studio will have a timing impact and is more commonly used for windo developers.
- Licensing Costs: Visual Studio may have some licensing cost, but most are free and open source.

<u>Mobile Devices:</u>

Server Side:

- Platform Characteristics: Portable, Flexible, and trackable.
- Platform Advantages: transportable and convenient. Better compatibility and cost effectiveness. There is a vast or a large userbase, almost everyone owns a mobile device,

so they are commonly used around the world. Mobile devices have a wider reach and better compatibility.

- Platform Weaknesses: It is a smaller display, and a mobile device may have a different view other than the real site and some of the older versions my lack support. The mobile device options can also be limited and highly sensitive with poor security, so it is more difficult to implement than other devices.
- Server-Based Deployment: The back end may be hosted on a server but is not a server-based deployment
- Potential Licensing Costs: Moderate

Client Side:

- Cost: Minimum
- Time: Maximum
- Expertise: Maximum

Developing Tools:

- Programming Languages: Java, HTML/CSS/JavaScript.
- IDEs and Tools:
    - Development: Android Studio, Visual Studio, and WebStorm.
- Impact: The developer needs experience in Java for Android development.
- Licensing Costs: Android Studio is free to use.

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**:
   A good Operating platform to start off with might be Windows. It seems to be the most versatile with the least complications. With multiple selections of software programs that are available that macOS can't even compete against. Windows is also very commonly used so most people have some experience with using the front end of windows. Windows also has consistent support and keeps up with updates which is important to address security issues to make windows less vulnerable and replaces outdated features with new to help improve software stability. It may also be easier to cross platforms since it is very similar to the other operating platforms. Microsoft has made an effort to be a first-class gaming platform by giving more access to hardware components and optimizing the performance of resource-intensive applications to specifically help developers. With all the resources that are given they make it too easy to not start developing DRAW IT OR LOSE IT with Windows.

2. **Operating Systems Architectures**:

Since we are creating a game for all devices, we will need to use a web-based platform. The most commonly used platforms are Java, CSS, and HTML because of the cross-platform tools and can be used on multiple devices. So, platforms such as macOS, Linux and windows would work appropriately and cross-platform IDE such as IntelliJ IDEA works on all three of those operating systems. Windows architecture consists of layers divided by two main components, the user mode and the kernel mode. The user mode is created for the interaction within the application and subsystems. Where the kernel mode is the core of the operating system that holds the hardware, drivers, and executive services of the development. X86 is also a commonly used instruction set architecture that was developed by Intel.

3. **Storage Management**:

Windows offers features such as the cloud server which provides plenty of storage so no worrying about running out of space. The cloud server is also very reliable and available across multiple platforms. Depending on the version of Windows that you are using you might also have access to a feature called storage sense which allows you to manage files on your hard drive to manage the space that may be taken up.

4. **Memory Management**:

The larger the game the more memory needed to store within it such as the library and database with all the pictures that are in the game. Memory can be managed by reducing the size of any images or videos and reducing the code for the program as short and sweet as possible. Windows memory manager monitors the behavior throughout the process. Windows also provides memory diagnostics tools that check and update the user if the memory is starting to run low.

5. **Distributed Systems and Networks**:

Although most of the operation platforms can be pretty similar there can be some complications or common errors that can result in communication issues, failed components, lagging performance, and connection problems. But it isn't too difficult to correct these situations, for example: using a load balancing system to distribute network traffic across the different servers.

6. **Security**:

In the window settings the user can control the secure data that is going in and out of the system. Using authorization to ensure that the modifications were not created without the user consent. There is provided security software that can be downloaded for mor protection, but windows also provide built-in anti-spyware solutions. In addition to regular security checks and updates each windows machine holds their very own VPN for more privacy and security.