CS 230 Project One Guidelines and Rubric

Competency

In this project, you will demonstrate your mastery of the following competency:

- Utilize software design templates and patterns to efficiently solve a problem

Scenario

You work for Creative Technology Solutions (CTS) as a Technology Consultant. CTS has recently taken on a new client, The Gaming Room. The Gaming Room wants to develop a web-based game that serves multiple platforms based on their current game, Draw It or Lose It, which is currently available in an Android app only.

Draw It or Lose It is loosely similar to the 1980s television game *Win, Lose or Draw*, where teams compete to guess what is being drawn. Rather than a player drawing images on an easel to help team members guess the puzzle (a phrase, title, or thing), the application will render images from a large library of stock drawings as clues. A game consists of four rounds of play lasting one minute each. Drawings are rendered at a steady rate and are fully complete at the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.

The staff at The Gaming Room does not know how to set up the environment. To facilitate the development of the web-based version of the gaming app, they will need your help in streamlining the development. You have been asked to prepare a software design document and begin developing the game application, addressing their software requirements. Keep in mind that the hardware requirements will come later as a result of the software application decisions. Your technical manager will want to review your progress with this client.

Specifically, the client has requested that the following **software requirements** be met for the game application:

- A game will have the ability to have one or more teams involved.
- Each team will have multiple players assigned to it.
- Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
- Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

Directions

You have been asked to prepare a software design document and begin developing the game application, addressing their software requirements.

Part I. Software Design Document: Use the template provided in the Supporting Materials section below to create a software design document for your client. Keep in mind your audience and be sure to communicate clearly and concisely in the template. You have been asked to complete the following sections of the template:

- **Executive Summary: Write a summary to introduce the software design problem and present a solution**. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.
- **Design Constraints: Identify the design constraints** for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.

The UML for the application has been previously created. Refer to the Supporting Materials section below to review what has been done, and continue to build upon the code to meet your client's software requirements.

- **Domain Model:** Review the UML diagram for the game application that represents the Domain Model within the software design template. Consider what information you have at this point regarding the game application and how it is represented in the model. Note: The Entity class is a base class introduced to hold common attributes and behaviors. **Describe the UML class diagram and explain how the classes relate to each other. Be sure to identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.**

Part II. Java Application: Use the code you submitted in the Project One Milestone to continue developing the game application in this project. Be sure to correct errors and incorporate feedback before submitting Project One.

Please note: The starter code for this project was provided in the Project One Milestone. If you have not completed the Project One Milestone, you will not be penalized in this assignment, but you will have additional steps to complete to ensure you meet all the components of Project One.

Review the class files provided and complete the following tasks to create a functional game application that meets your client's requirements. You will submit the completed game application code for review.

Begin by reviewing the base Entity class. It contains the attributes 'id' and 'name', implying that all entities in the application will have an identifier and name.

- **Software Design Patterns:** Review the GameService class. Notice the static variables holding the next identifier to be assigned for game id, team id, and player id.

    - Referring back to the Project One Milestone, be sure that you **use the singleton pattern to adapt an ordinary class**, so only one instance of the GameService class can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.
    - Your client has requested that the game and team names be unique to allow users to check whether a name is in use when choosing a team name. Referring back to the Project One Milestone, be sure that you **use the iterator pattern to complete the addGame() and getGame() methods**.
    - **Create a base class called Entity**. The Entity class must hold the common attributes and behaviors (as shown in the UML diagram provided in the Supporting Materials section below).
    - **Refactor the Game class to inherit from this new Entity class**.
    - **Complete the code for the Player and Team classes**. Each class must derive from the Entity class, as demonstrated in the UML diagram.
    - Every team and player must have a unique name by searching for the supplied name prior to adding the new instance. **Use the iterator pattern in the addTeam() and addPlayer() methods.**

- **Functionality and Best Practices**
    - Once you are finished coding, use the main() method provided in the ProgramDriver class to run and test the game application to ensure it is functioning properly.
    - Be sure your code **demonstrates industry standard best practices to enhance the readability of your code, including appropriate naming conventions and in-line comments** that describe the functionality.

What to Submit

To complete this project, you must submit the following:

Part I. Software Design Template Word Document

Complete the software design document using the template provided. The document must address each section outlined in the directions provided above.

Part II. Java Application

Submit the completed game application code. This will be the first prototype showcasing how The Gaming Room can offer their game, Draw It or Lose It, in a web-based format and serve multiple platforms. Be sure to download and save the project as CS 230 Project One Game App. Compress your Eclipse project directory into a single ZIP file to be submitted.

Supporting Materials

The following resource(s) may help support your work on the project:

Project One Milestone Guidelines and Rubric

Use the diagram and code you submitted in the Project One Milestone to continue developing the game application in this project. Be sure to correct errors and incorporate feedback before submitting Project One.

If you have not completed the required Project One Milestone assignment from Module Two, you will have additional steps to complete to ensure you meet all the components for Project One. The starter code for this project was provided in the Project One Milestone Prompt.

Project One UML Diagram

Review the completed UML diagram for the game application. A text version Word Document of this diagram is available.

Uploading Files to Eclipse Desktop Version Tutorial PDF

You will write, test, and run the Java application in Eclipse. Review this tutorial to learn how to upload the zipped project file into the IDE.

Downloading and Zipping Files in Eclipse Tutorial PDF

Review this tutorial to learn how to download your Java application from Eclipse as a zipped file folder for submission.

Project One Rubric

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Software Design Template: Executive Summary	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Summarizes the problem and presents a solution that includes critical information outlining how to proceed with the solution	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include missing pertinent details or rationale for solution	Does not attempt criterion	15
Software Design Template: Design Constraints	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Identifies the appropriate design constraints of successfully setting up a web-based distributed environment	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include inclusion of all design constraints related to software requirements provided	Does not attempt criterion	20
Software Design Template: Domain Model	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Describes how classes relate to one another, identifying any object-oriented programming principles illustrated in a class diagram	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include omission of specific object-oriented principles	Does not attempt criterion	20

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Java Application: Software Design Patterns	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Uses software design patterns to efficiently meet software application requirements	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include successful implementation of patterns	Does not attempt criterion	25
Java Application: Program Functionality and Best Practices	Exceeds proficiency by demonstrating industry standard best practices, including in-line comments and appropriate naming conventions to enhance readability of code	Develops functional Java code that illustrates software design pattern approach including multiple classes	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include implementation of software design patterns in code	Does not attempt criterion	15
Articulation of Response	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose	Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability	Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas	5
				Total:	100%