PROJECT: THE TB QUEST GAME S1 (INTRODUCTION, GAME FRAMEWORK, GAME SCREEN, MENUS, AND PLAYER SETUP)

OVERVIEW

Sprint 1 will include developing the basic structure and a simple UI for the application. This sprint will require the initial development of the Player class including appropriate fields, properties, methods, and constructors.

REQUIREMENTS

Include the following

- Game Flow: Splash Screen, Intro Screen, Player Setup, Menu, and Closing Screen
- Customized Theme
- Functionality: Player Setup and Edit

SUBMIT FOR GRADE

1. Prepare for submission.
   a. Download and review the **Sprint 1 Features and Requirements Checklist** to ensure you have all requirements met. Indicate all missing requirements by highlighting each using MS Word.
   b. Create a GitHub repository.
   c. Push the most current version of the solution to the GitHub repository.
   d. Create a 3-5 minute video presentation demonstrating all of the features listed in the **Sprint 1 Features and Requirements Checklist**.
2. Login to Moodle and open the ***Project: The TB QUEST Game (Sprint 1)*** assignment. **Note**: Submissions will not be graded without all items below completed.
   a. Submit the link to the streaming video presentation. Be sure to follow the **Video Presentation Specifications** document located in **Course Resources** on Moodle.
   b. Submit the link to the remote repository.
   c. Submit the completed **Sprint 1 Features and Requirements Checklist.**
3. Return to the Moodle assignment later to view your grade.

PROJECT: TB QUEST GAME SPRINT 1 – MARKING GUIDE

Conventions, Readability, and Structure	• Solution, project and folder structures adhere to the standards for a given design pattern. • Files, classes and their elements adhere to the course naming conventions. • File, class, and class element names are descriptive and consistent. • Whitespace is used effectively and consistently. Nested elements are indented and code blocks are separated by blank lines. • Classes, constructors and methods use XML tag commenting including parameter and return elements. • Significant code blocks use single line commenting. • Methods perform a specific, single action. • Code blocks are not duplicated in more than one location. • Inheritance and polymorphism are used to organize classes. • Separation of concerns, modularity and reusability are taken into consideration • The code does not include unused or extraneous elements.	**10**	
Robustness	• The UI is clear regarding the type of user interaction and input, and provides guidance when an exception is generated by the user. • All potential exceptions are trapped and handled.	**10**	
Features – Level I	• All features and requirements are included	**50**	
Features – Level II	• All features and requirements are included	**10**	
Features – Level III	• All features and requirements are included	**10**	
Overall Quality, Creativity & Effort	• The application feels complete with attention paid to details. • The application demonstrates creativity on the part of the developer. • The application demonstrates significant effort on the part of the developer. Optional enhancements are included.	**10**	
	Final Grade	**100**	

PROJECT: TB QUEST GAME SPRINT 1 - FEATURES AND REQUIREMENTS CHECKLIST

Note: The class and class member names are generic unless in bolded italics in the Level Requirements. The student is required to modify all class names unless in bolded italics to be consistent with their chosen theme.

1. Complete the checklist below. Provide any additional comments in the space below the checklist.
2. Self-score in the provided area at the bottom of the checklist.

	Level I	**Level II (include all Level I requirements)**	**Level III (include all Level II requirements)**
Theme	☐ A consistent, custom theme, different than the demo, is implement		
Character Class	☐ Base class for player class ☐ Include string, int, and bool fields	☐ Include enum field	☐ Virtual method
Player Class	☐ Has a name consistent with theme ☐ Derived from ***Character*** class ☐ Include string, int, and bool fields beyond the ***Character*** class	☐ Include enum field beyond the ***Character*** class	☐ Additional fields ☐ Overloaded method
Game Flow	☐ Splash Screen ☐ Intro Screen ☐ Player Setup ☐ Menu ☐ Closing Screen		
Player Actions	☐ Player Info ☐ Exit		☐ Player Info ☐ Player Edit ☐ Exit
Robustness and Validation	☐ No user input is validated	☐ Most user input is validated	☐ All user input is validated ☐ Game is "bomb-proof"
.NET and OOP Concepts and Elements Applied	☐ MVC used	☐ MVC used consistently	☐ Inheritance; virtual and overloaded methods
Marking Value	**50 Points**	**10 Points**	**10 Points**
Self-Score			