PROJECT: THE TB QUEST GAME S2 (GAME MAP, LOCATIONS, TRAVEL, AND ACCESS)

OVERVIEW

Sprint 2 will include the development of the *GameMap* and *Location* classes along with implementing player travel, look around, and visited locations functionality. Limiting access to locations based on current location and game events will be discussed.

REQUIREMENTS

Include the following

- Functionality: Travel, Look Around, and Visited Locations
- GameMap and Location classes
- Location accessibility management
- Game map locations with names, descriptions,

SUBMIT FOR GRADE

1. Prepare for submission.
    a. Download and review the **TB Quest Game Sprint 2 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 S2*** 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 **TB Quest Game Sprint 2 Checklist.**
3. Return to the Moodle assignment later to view your grade.

PROJECT: TB QUEST GAME SPRINT 2 - 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	☐ All displayed text is modified from the demo to be consistent with the new theme		
Character Class	☐ Base class for player class ☐ Include string, int, and bool fields ☐ Include current location id	☐ Include enum field	
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
Map Class	☐ Include a list of locations ☐ All locations accessible from all locations	☐ Location accessibility determined by Boolean in location class	☐ Location accessibility determined by a more complex method
Location Class	☐ Includes a name and significant description ☐ 4 locations	☐ 8 locations ☐ Limited accessibility	☐ Includes experience points
Game Flow	☐ Splash Screen ☐ Intro Screen ☐ Player Setup ☐ Menu ☐ Closing Screen		
Game Play			☐ Track experience points for visiting locations and update Game Status box

Player Actions	☐ Player Info ☐ Look Around ☐ Travel ☐ List All Locations ☐ Exit		☐ Player Info ☐ Player Edit ☐ Look Around ☐ Travel ☐ Locations Visited ☐ 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	☐ Encapsulation well implemented
Marking Value	**10 Points**	**5 Points**	**5 Points**
Self-Score			