PROJECT: THE TB QUEST GAME (SPRINT 2 – ITEMS, TREASURES, ITEMS LIST, AND TREASURES LIST)

OVERVIEW

The second sprint will develop the GameObject base class and classes derived from it such as the Item class and the Treasure class. The player's interaction with each game object will be implemented

Some possible player and game item interactions would include.

- Adding or removing an item from the player's inventory
- Adding to or spending the player's treasure
- Receiving information from an item
- Receiving experience points, health, and/or lives from an item
- Transporting the player to a new location

Note: Game objects may be designed such that they are either fixed in a given location or are mobile, i.e. the player may pick them up or put them down as they move through the game.

SUBMIT FOR GRADE

1. Prepare for submission.
   1.1. Run the application to confirm that all included features are fully functional and tested. Any features that are not fully functional MUST be noted on the **Sprint 2 Features and Requirements Checklist** and commented out in the code. Remember, a sprint deliverable should be a robust solution and able to be run by the stakeholders without any issues per the task list.
   1.2. Download and complete the **Sprint 2 Features and Requirements Checklist** per the instructions at the top of the checklist. The checklist is a Word document so you have two options to prepare it for submission.
      1.2.1. Print the document, complete it, and then scan it.
      1.2.2. Open the document in Word, highlight the incomplete items in yellow, and then save it.
   1.3. Create a 3-5 minute video presentation demonstrating all of the features listed in the **Sprint 2 Features and Requirements Checklist**. (Suggested App: Jing)
   1.4. Push the most current version of the solution to GitHub.
2. Login to Moodle and open the ***Project: The TB QUEST Game (Sprint 2)*** assignment. Note: Submissions will not be graded without all items below completed.
   2.1. Submit the link to the streaming video presentation.
   2.2. Submit the link to the remote repository.
   2.3. Submit the **Sprint 2 Features and Requirements Checklist.**
1. Return to the Moodle assignment later to view your grade.

SPRINT 2 - FEATURES AND REQUIREMENTS CHECKLIST

Note: To earn Sprint 2 level points, all Sprint 1 requirements for the same level must also be implemented.

> **Note**: The class and class member names are generic unless in bolded italics in the Level Requirements. The student is required to modify the names 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	☑ The theme is apparent and consistent		
Locations	☐ Some locations have an item or treasure	☐ All locations have an item or treasure	☑ Some locations have multiple items and/or treasure
GameObject Class		☐ Virtual Class GameObjectID Name Description SpaceTimeLocationID	☑ Abstract Class GameObjectID Name Description SpaceTimeLocationID HasValue Value CanAddToInventory InInventory
Item Class	☐ Unique, non-derived class	☐ Inherits from the GameObject class	☑ Inherits from the GameObject class
Treasure Class	☐ Unique, non-derived class	☐ Inherits from the GameObject class	☑ Inherits from the GameObject class
Player Actions	☐ Look Around ☐ Look At ☐ Pick Up Item ☐ Put down Item ☐ Travel ☐ Display All Destinations ☐ Player Info ☐ Player Inventory ☐ Exit	☐ Look Around ☐ Look At ☐ Pick Up Item ☐ Pick Up Treasure ☐ Put down Item ☐ Put down Treasure ☐ Travel ☐ Display All Destinations ☐ Player Info ☐ Player Inventory ☐ Player Treasure ☐ Exit	☐ All Level II functionality ☐ Additional functionality when the player interacts with an object such as teleportation, added experience points, magical capabilities, etc.

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 Elements Applied	☐ Inheritance; virtual and override methods	☐ Abstract classes and methods	☐ Dictionary
Marking Value	**10 Points**	**5 Points**	**5 Points**
Self-Score			