PROJECT: THE TB QUEST GAME (SPRINT 4 – NPC'S)

OVERVIEW

The forth sprint will develop the Traveler's interactions with the game including interacting with other characters (NPCs), derived from the Character class. The sprint will also feature more advanced interactions and game play.

Some possible player and game item interactions would include.

- Talking with an NPC player to get information
- Exchanging items and treasures with an NPC
- Receiving experience points, health, and/or lives from an NPC
- Transporting the player to a new location

PROJECT FEATURES AND TASK LIST

Sprint 4 deliverables will include the following new features and requirements.

- A minimum of one NPC class derived from the Character class
- A student defined interface that is implemented with one of the NPC classes
- Player interaction with an NPC

INSTRUCTIONS

1. Develop the application to meet the specification in the **Features and Requirements Checklist** and **Delivery Sheet**.

SUBMIT FOR GRADE

1. Prepare for submission.
    a. Test the application to confirm that all included features are fully functional and tested. Any non-functional feature must be disabled and commented out. Remember, a sprint deliverable should be a robust solution and able to be run by the stakeholders without any issues per the task list.
    b. Download and complete the **Sprint 4 Features and Requirements Checklist** and **Sprint 4 Delivery Sheet** per the instructions at the top of each document. Any features that are not fully functional MUST be noted on the **Sprint 4 Delivery Sheet** and commented out in the code. The documents are Word documents so you have two options to prepare it for submission.
        i. Print the documents, complete them, and then scan them.
        ii. Open the document in Word, highlight the incomplete items in yellow, and then save it.
    c. Create a 3-5 minute video presentation that follows the guidelines of the **Video Project Specifications** document in **Course Resources**. Demonstrate all of the features listed in the **Sprint 3 Features and Requirements Checklist**. (Suggested App: Jing)
    d. Push the most current version of the solution to GitHub.
2. Login to Moodle and open the ***Project: TB QUEST Game (Sprint 4)*** 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 **Sprint 4 Features and Requirements Checklist** and **Sprint 4 Delivery Sheet**
3. Return to the Moodle assignment later to view your grade.

PROJECT: TB QUEST GAME SPRINT 2 – 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 4 - FEATURES AND REQUIREMENTS CHECKLIST

Note: To earn Sprint 4 level points, all Sprint 3 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
NPC Class	☐ Unique, non-derived class	☐ Inherits from the Character class	☐ Implements an interface
Player Actions	☐ Look Around ☐ Look At ☐ Pick Up Item ☐ Put down Item ☐ Talk To ☐ Travel ☐ Locations Visited ☐ List all Locations ☐ List All Game Objects ☐ List All NPCs ☐ Player Info ☐ Player Inventory ☐ Exit	☐ Look Around ☐ Look At ☐ Pick Up Item ☐ Pick Up Treasure ☐ Put down Item ☐ Put down Treasure ☐ Talk To ☐ Travel ☐ Locations Visited ☐ List all Locations ☐ List All Game Objects ☐ List All NPCs ☐ 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. ☐ Additional functionality when the player interacts with an NPC such as attack, defend buy, sell, give, 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 ☐ Interface
Marking Value	**50 Points**	**10 Points**	**10 Points**
Self-Score			

PROJECT: TB QUEST GAME SPRINT 4 – DELIVERY SHEET

Complete the following document and submit to Moodle.

1. Provide any comments necessary to explain why a feature/requirement may have deviated from the requirements.
2. Self-score each feature or requirement.
3. Total the self-score points.
4. Save and submit to Moodle.

	Requirements	Student Comment	Points Possible	Self Score
Conventions	☐ 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.		**5**	
Readability	☐ 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.		**5**	
Structure	☐ Class elements are organized and adhere to the course convention. ☐ 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.		**5**	
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.		**5**	

Efficiency & Elegance	☐ Appropriate algorithms are used and the code is small, yet still readable. ☐ The code does not include unused or extraneous elements.		**5**	
Level I	☐ All features and requirements are implemented.		**50**	
Level II	☐ All features and requirements are implemented.		**10**	
Level III	☐ All features and requirements are implemented.		**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.		**5**	
		Total	**100**	