

PROJECT: THE TB QUEST GAME S3 (GAME OBJECTS)

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.
 - a. 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 3 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.
 - b. Download and complete the **Sprint 3 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.
 - i. Print the document, complete it, and then scan it.
 - ii. Open the document in Word, highlight the incomplete items in yellow, and then save it.
 - c. Create a 3-5 minute video presentation demonstrating all of the features listed in the **Sprint 2 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: The TB QUEST Game (Sprint 3)** 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 3 Features and Requirements Checklist**.
3. Return to the Moodle assignment later to view your grade.

PROJECT: TB QUEST GAME SPRINT 3 - FEATURES AND REQUIREMENTS CHECKLIST

Note: To earn Sprint 3 level points, all Sprint 2 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	<input type="checkbox"/> The theme is apparent and consistent		
Locations	<input type="checkbox"/> Some locations have an item or treasure	<input type="checkbox"/> All locations have an item or treasure	<input type="checkbox"/> Some locations have multiple items and/or treasure
GameObject Class		<input type="checkbox"/> Virtual Class GameObjectID Name Description SpaceTimeLocationID	<input type="checkbox"/> Abstract Class GameObjectID Name Description SpaceTimeLocationID HasValue Value CanAddToInventory InInventory
Item Class	<input type="checkbox"/> Unique, non-derived class	<input type="checkbox"/> Inherits from the GameObject class	<input type="checkbox"/> Inherits from the GameObject class
Treasure Class	<input type="checkbox"/> Unique, non-derived class	<input type="checkbox"/> Inherits from the GameObject class	<input type="checkbox"/> Inherits from the GameObject class
Player Actions	<input type="checkbox"/> Look Around <input type="checkbox"/> Look At <input type="checkbox"/> Pick Up Item <input type="checkbox"/> Put down Item <input type="checkbox"/> Travel <input type="checkbox"/> Locations Visited <input type="checkbox"/> List All Locations <input type="checkbox"/> List All Game Objects <input type="checkbox"/> Player Info <input type="checkbox"/> Player Inventory <input type="checkbox"/> Exit	<input type="checkbox"/> Look Around <input type="checkbox"/> Look At <input type="checkbox"/> Pick Up Item <input type="checkbox"/> Pick Up Treasure <input type="checkbox"/> Put down Item <input type="checkbox"/> Put down Treasure <input type="checkbox"/> Travel <input type="checkbox"/> Locations Visited <input type="checkbox"/> List All Locations <input type="checkbox"/> List All Game Objects <input type="checkbox"/> Player Info <input type="checkbox"/> Player Inventory <input type="checkbox"/> Player Treasure <input type="checkbox"/> Exit	<input type="checkbox"/> All Level II functionality <input type="checkbox"/> Additional functionality when the player interacts with an object such as teleportation, added experience points, magical capabilities, etc.

Robustness and Validation	<input type="checkbox"/> No user input is validated	<input type="checkbox"/> Most user input is validated	<input type="checkbox"/> All user input is validated <input type="checkbox"/> Game is "bomb-proof"
.NET and OOP Elements Applied	<input type="checkbox"/> Inheritance; virtual and override methods	<input type="checkbox"/> Abstract classes and methods	<input type="checkbox"/> Dictionary
Marking Value	10 Points	5 Points	5 Points
Self-Score			

PROJECT: TB QUEST GAME SPRINT 3 – 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	<input type="checkbox"/> Solution, project and folder structures adhere to the standards for a given design pattern. <input type="checkbox"/> Files, classes and their elements adhere to the course naming conventions.		5	
Readability	<input type="checkbox"/> File, class, and class element names are descriptive and consistent. <input type="checkbox"/> Whitespace is used effectively and consistently. Nested elements are indented and code blocks are separated by blank lines. <input type="checkbox"/> Classes, constructors and methods use XML tag commenting including parameter and return elements. <input type="checkbox"/> Significant code blocks use single line commenting.		5	
Structure	<input type="checkbox"/> Class elements are organized and adhere to the course convention. <input type="checkbox"/> Methods perform a specific, single action. <input type="checkbox"/> Code blocks are not duplicated in more than one location. <input type="checkbox"/> Inheritance and polymorphism are used to organize classes. <input type="checkbox"/> Separation of concerns, modularity and reusability are taken into consideration.		5	
Robustness	<input type="checkbox"/> The UI is clear regarding the type of user interaction and input, and provides guidance when an exception is generated by the user. <input type="checkbox"/> All potential exceptions are trapped and handled.		5	

Efficiency & Elegance	<input type="checkbox"/> Appropriate algorithms are used and the code is small, yet still readable. <input type="checkbox"/> The code does not include unused or extraneous elements.		5	
Level I	<input type="checkbox"/> All features and requirements are implemented.		50	
Level II	<input type="checkbox"/> All features and requirements are implemented.		10	
Level III	<input type="checkbox"/> All features and requirements are implemented.		10	
Overall Quality, Creativity & Effort	<input type="checkbox"/> The application feels complete with attention paid to details. <input type="checkbox"/> The application demonstrates creativity on the part of the developer. <input type="checkbox"/> The application demonstrates significant effort on the part of the developer.		5	
		Total	100	