

Items Is Items

Software Design Document

Version: 3.0

Prepared by: Jordan Betcher, Job Betcher, Annick Stefanin

Adviser: Sara Farag

Bellevue College Computer Science

CS 481-483

3/15/2019

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms and Abbreviations	4
2. System Overview	4
2.1 Product Perspective	4
2.2 Product Functionality	5
2.2.1 Menu:	5
2.2.2 In-Game:	5
2.2.2.1 User:	5
2.2.2.2 NPC:	6
2.2.2.3 Items:	6
2.2.2.4 Item Effects:	6
2.2.2.5 GUI:	6
2.2.2.6 Quest:	6
2.3 Operating Environment	6
3. System Architecture	7
3.1 Architectural Design	8
3.2 Decomposition Description	7
3.2.1 Sequence Diagram	7
Item on Item:	8
3.2.2 Components	8
3.3 Dependency Description	8
3.3.1 TextMeshPro	8
Description	8
3.3.2 Combination Manager	8
Description	8
3.3.3 Quest States	9
Description	9
How to use Quest States	9
3.4 Interface Description	9
3.4.1 User Interfaces (GUI)	9
Quest Giver	10
4. Detailed Design	12
4.1 Class Diagrams	17
4.1.1 NPC Class:	12
4.1.2 Combination:	13

4.1.3 GameManager:	13
4.1.4 Validator	14
4.1.5 Effect	14
4.1.6 Quest:	15
4.1.7 Item:	16
4.2 Data Flow Diagram	17
5. RTM (Requirements Traceability Matrix)	18

1. Introduction

1.1 Purpose

The document will clarify how the different classes will interact with each other for the Items is Items game project. The Items is Items project is a 2D top down game designed around the use of items. The viewers of this document are programmers, and the instructors grading this capstone project, and players keen on learning the games inner workings of the game.

1.2 Scope

The scope will only cover custom scripts built in unity and how to use them.

1.3 Definitions, Acronyms and Abbreviations

Player - The user playing the game

NPC - Non-Player Character, which are controlled by the computer

AI - Artificial Intelligence is the brains behind the NPC directing what it does and how it will act

GUI - Graphical User Interface, or more simply, what the player will see

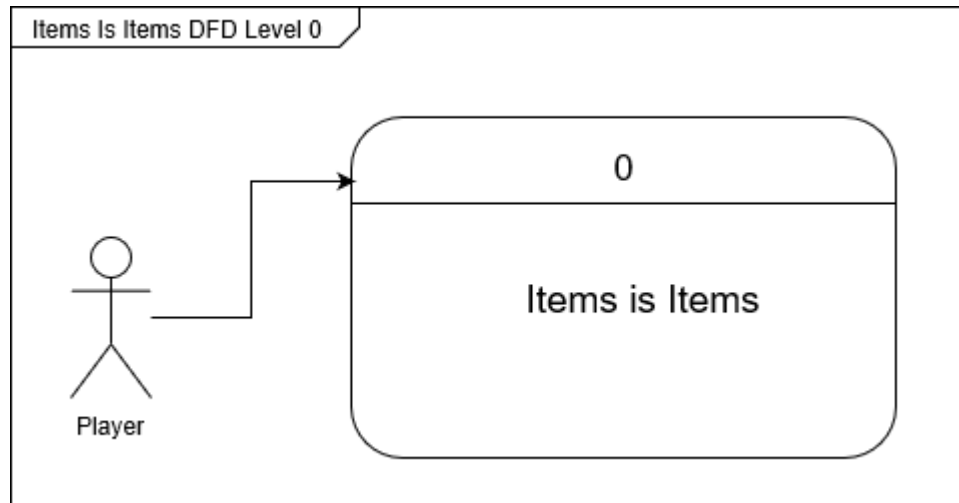
RTM - The Requirements Traceability Matrix, is the matrix connecting the various documents together

SRS - The Software Requirements Specifications document for this project

2. System Overview

2.1 Product Perspective

The game is built to provide buyers with a slow-paced, top-down item puzzle experience that is enjoyable to the user. It will be an enclosed system that run on a computer or tablet and requires the input of mouse or touch respectively. Using base classes in unity such as monobehaviour and scriptableobject.



The game is an enclosed system that consists of a menu screen allowing the player to manipulate world saves and the overworld in which the player traverses.

The menu greets the player upon opening the program. The menu allows the player to instantiate a new instance of a game or load a previously saved instance.

The overworld consist of the player and items that are separated into two different categories: NPCs and static item objects. Items can interact with each other, the player, or NPC objects. The player manipulates the world through movement and the use of items.

2.2 Product Functionality

2.2.1 Menu:

- User can start a new game
- User can load a previously saved game
- User can exit the program

2.2.2 In-Game:

2.2.2.1 User:

- User can move the character using left mouse click
- User can add items to inventory by running into them
- User can select items in inventory by left clicking them
- User can combine selected items by dropping selected item onto another item
- User can use item on another item by dropping selected item from inventory onto another item
- User can use item on self by dropping selected item from inventory onto self

- User can use item on NPC by dropping selected item from inventory onto NPC
- User can solve puzzles using items in their inventory
- User can receive physical reward upon quest completion
- User can receive progress on another quest upon quest completion
- User can receive progress on another quest upon item acquisition
- User drops all items in inventory on respawn

2.2.2.2 NPC:

- NPC can move towards items it is attracted to
- NPC can move towards other NPCs it is hostile to
- NPC can run away from other NPCs it is scared of
- NPC can return to their selected station
- NPC can use items it touches
- NPC can give quests to user through dialogue
- User can complete a quest by dropping the correct item onto them

2.2.2.3 Items:

- Items can combine with other items to produce a new item
- Items can be used to destroy another item
- Items can be used to affect another item
- Items can be used to affect a NPC
- Items can be used to affect the character

2.2.2.4 Item Effects:

- Items could destroy another item
- Item could change another item's color
- Item can teleport other item on touch

2.2.2.5 GUI:

- There is a button that restarts the level

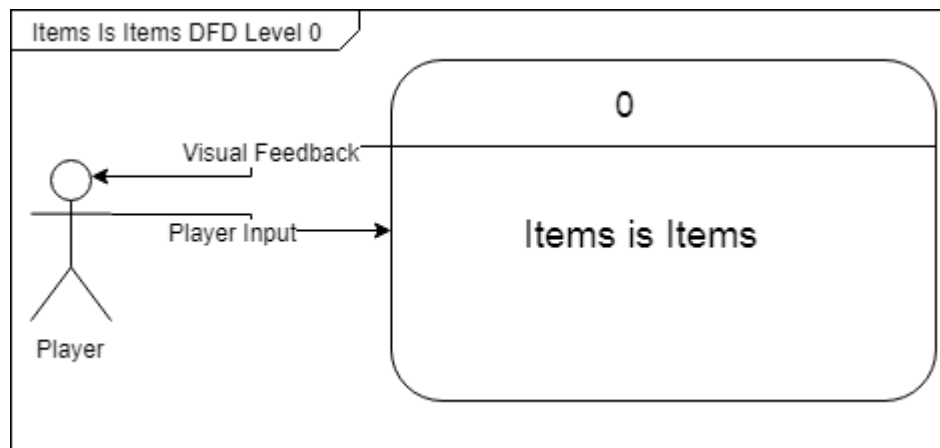
2.2.2.6 Quest:

- When Users clicks a quest giver a text box pops up showing the quest
- Giving the quest item moves the quest to the next state
- Giving the quest item can spawn a reward

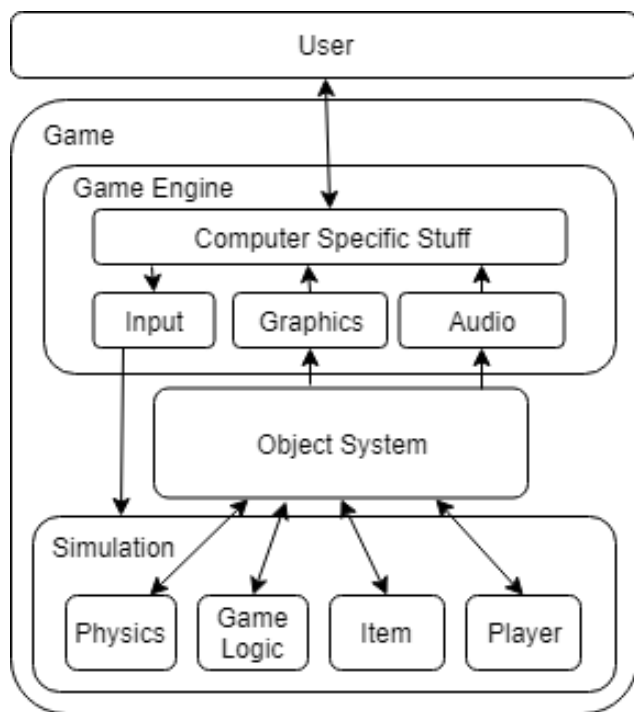
2.3 Operating Environment

Our software will be installed on user computer systems on Windows 10 Operating Systems, Mac Operating Systems, or Android Operating Systems. The software will require no outside connection.

3. System Architecture



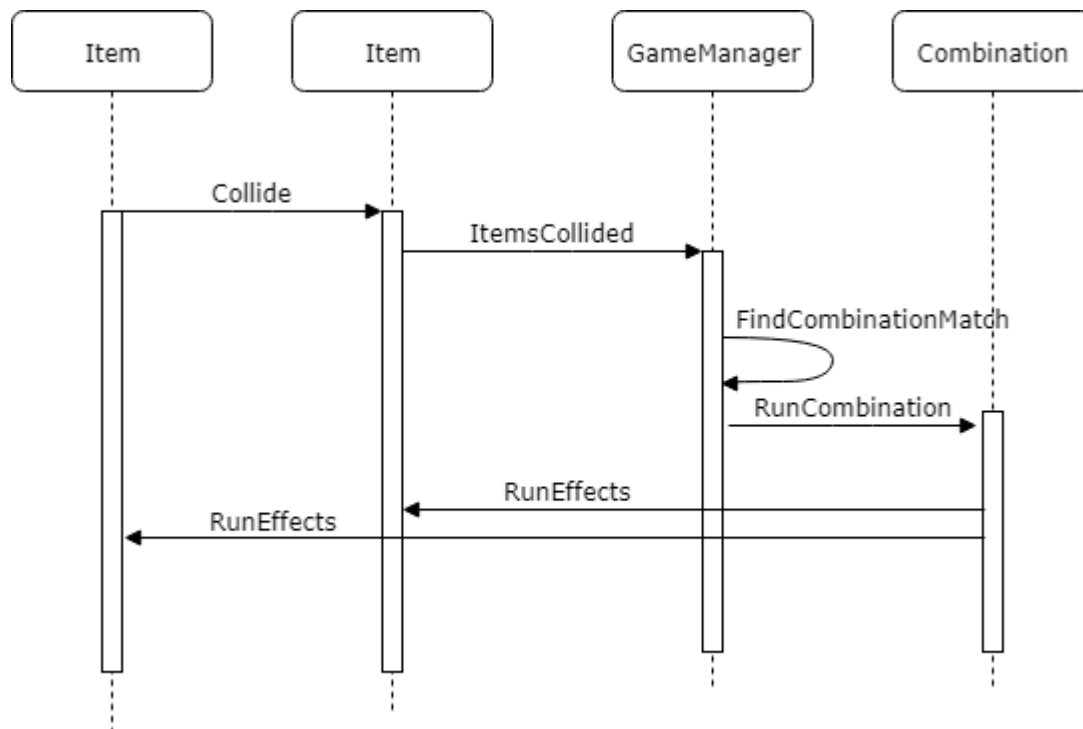
3.1 Architectural Design



3.2 Decomposition Description

3.2.1 Sequence Diagram

Item on Item:



3.2.2 Components

3.3 Dependency Description

3.3.1 TextMeshPro

Description

A unity provided text addon that increases the ease of use and flexibility of text.

3.3.2 Combination Manager

Description

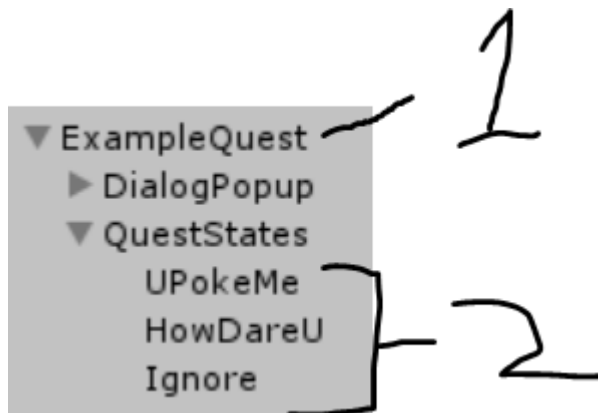
A manager for combinations that Job made. Allowing you to create combinations in the resources that are persistent across all scenes.

3.3.3 Quest States

Description

A set of reorderable states that you can attach to QuestGivers, created by Jordan.

How to use Quest States



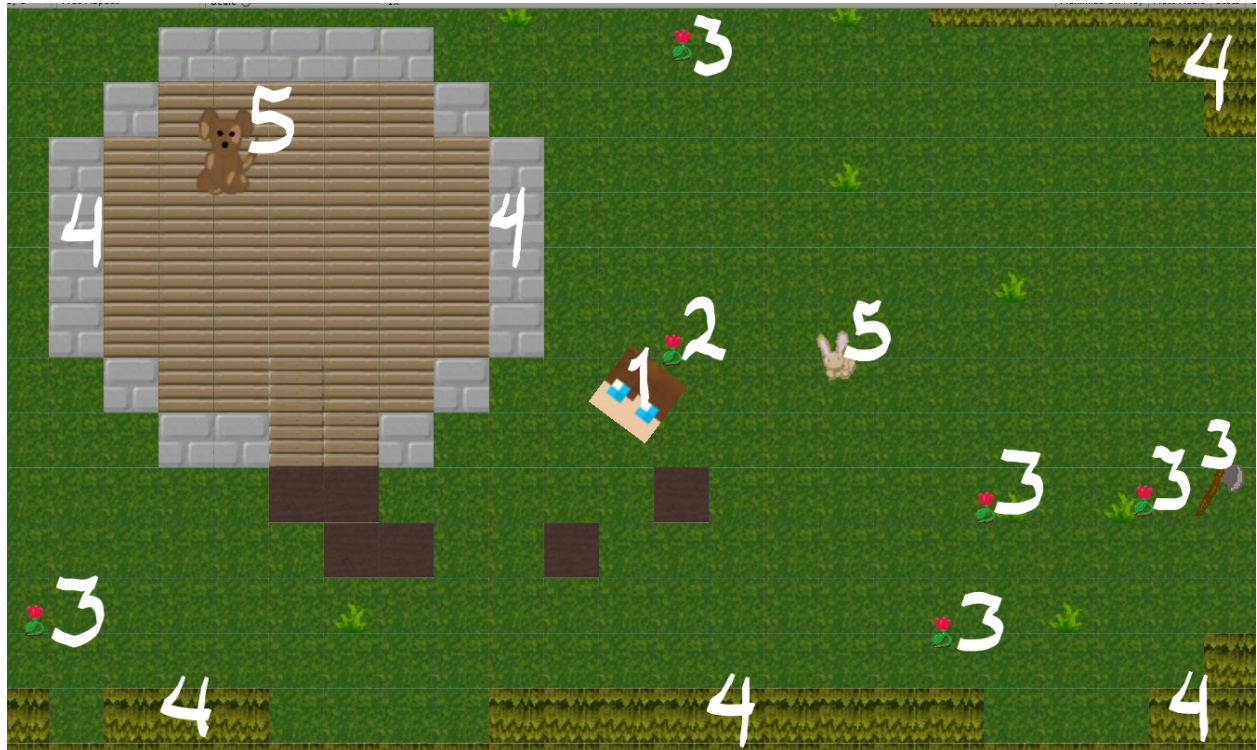
1. Is the object with the quest class or in other words, the quest giver
2. Is the quest states, you order them here to tell the quest in which order they go

3.4 Interface Description

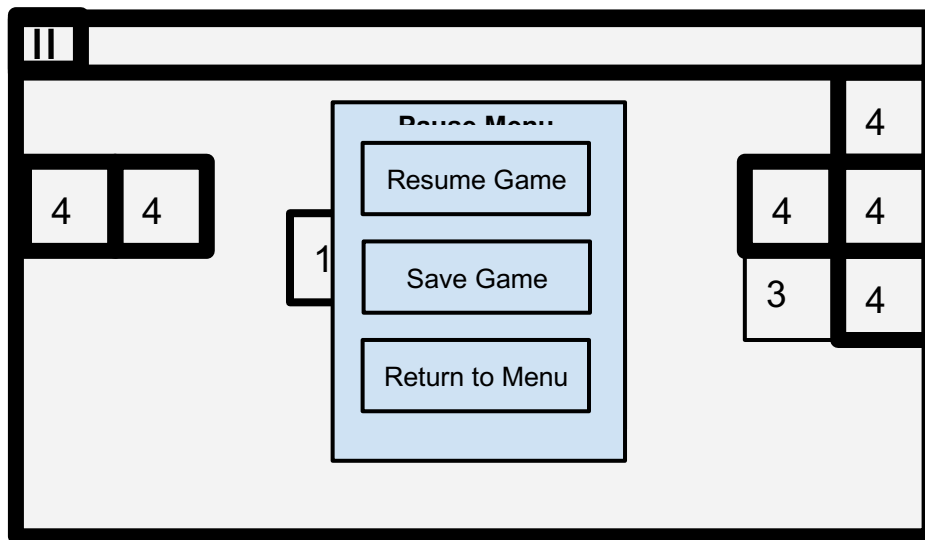
3.4.1 User Interfaces (GUI)

The user will left click to use the buttons. The user will left click to use select / use items. The user will right click to move.





- 1: Is the player
- 2: Is an item that follows behind the player
- 3: Are the items that the player hasn't picked up
- 4: Is the terrain that the player can't walk through
- 5: Are NPCs



Quest Giver

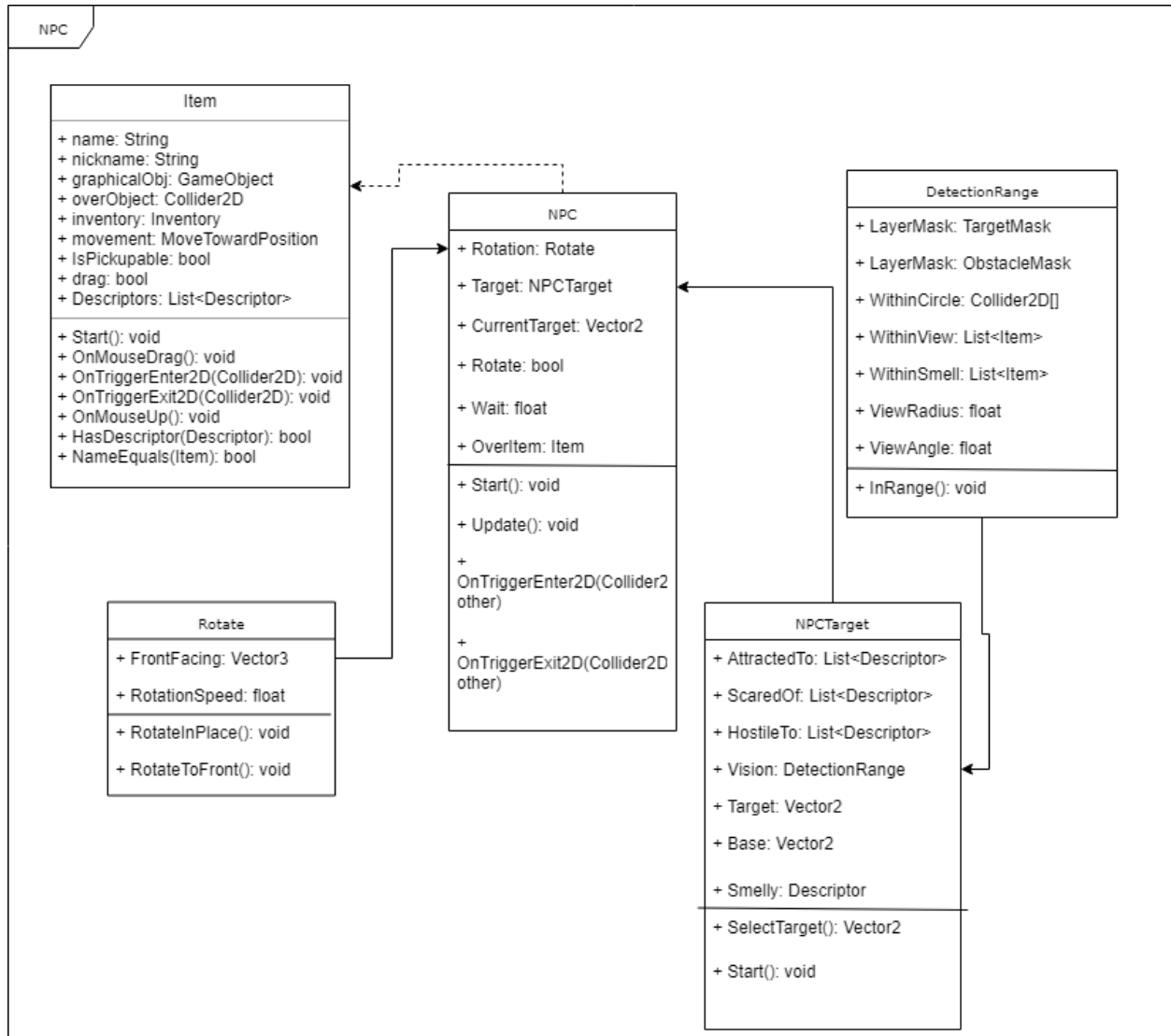


1. Is the quest giver
2. Is the Dialog that hints/tells about how to move on to the next part of the quest and the rewards for completing the quest

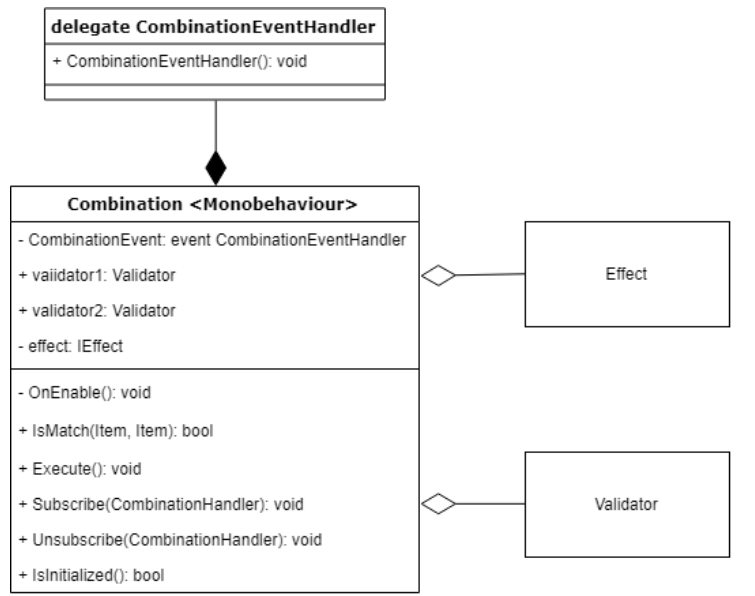
4. Detailed Design

4.1 Class Diagrams

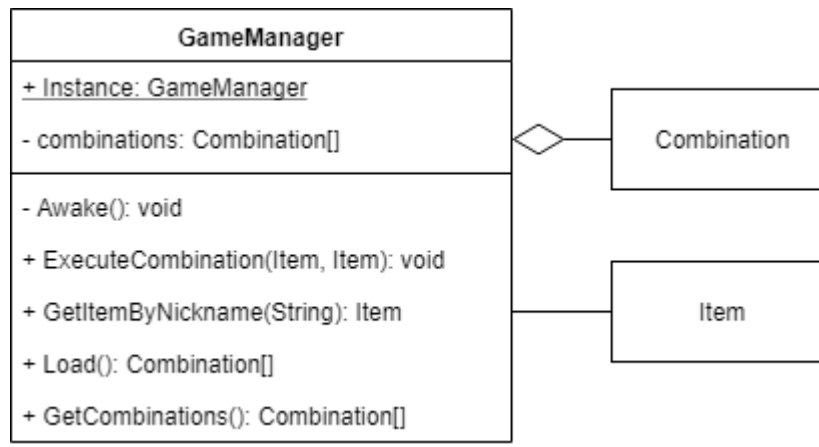
4.1.1 NPC Class:



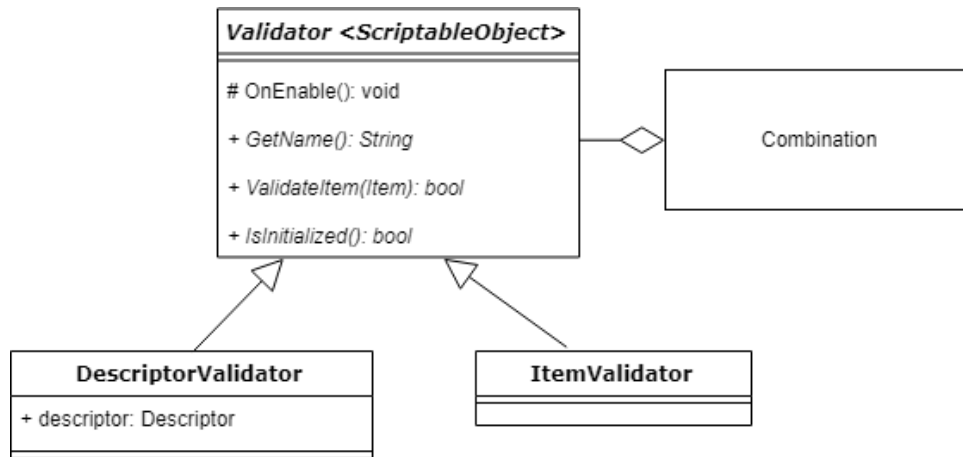
4.1.2 Combination:



4.1.3 GameManager:

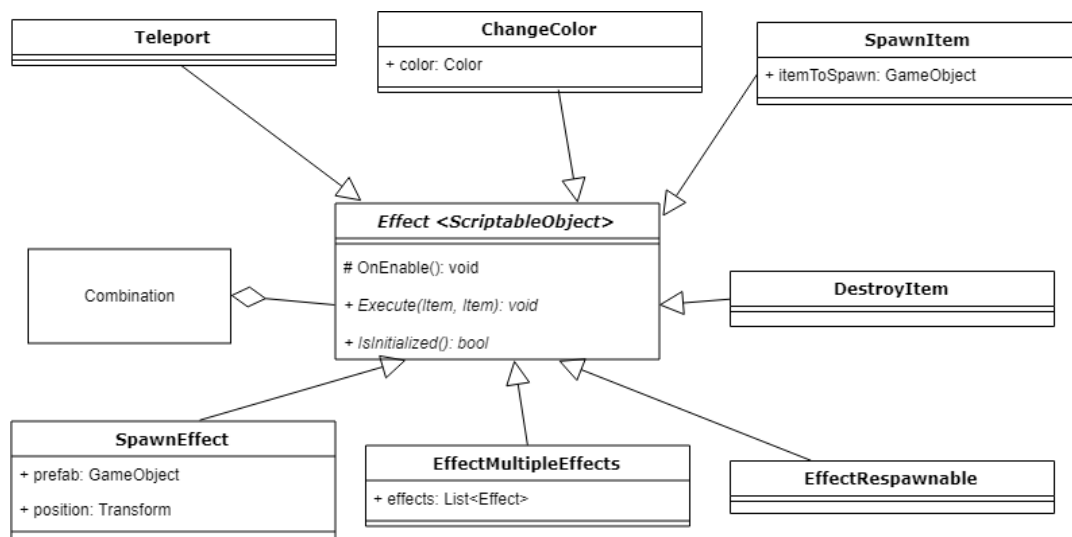


4.1.4 Validator



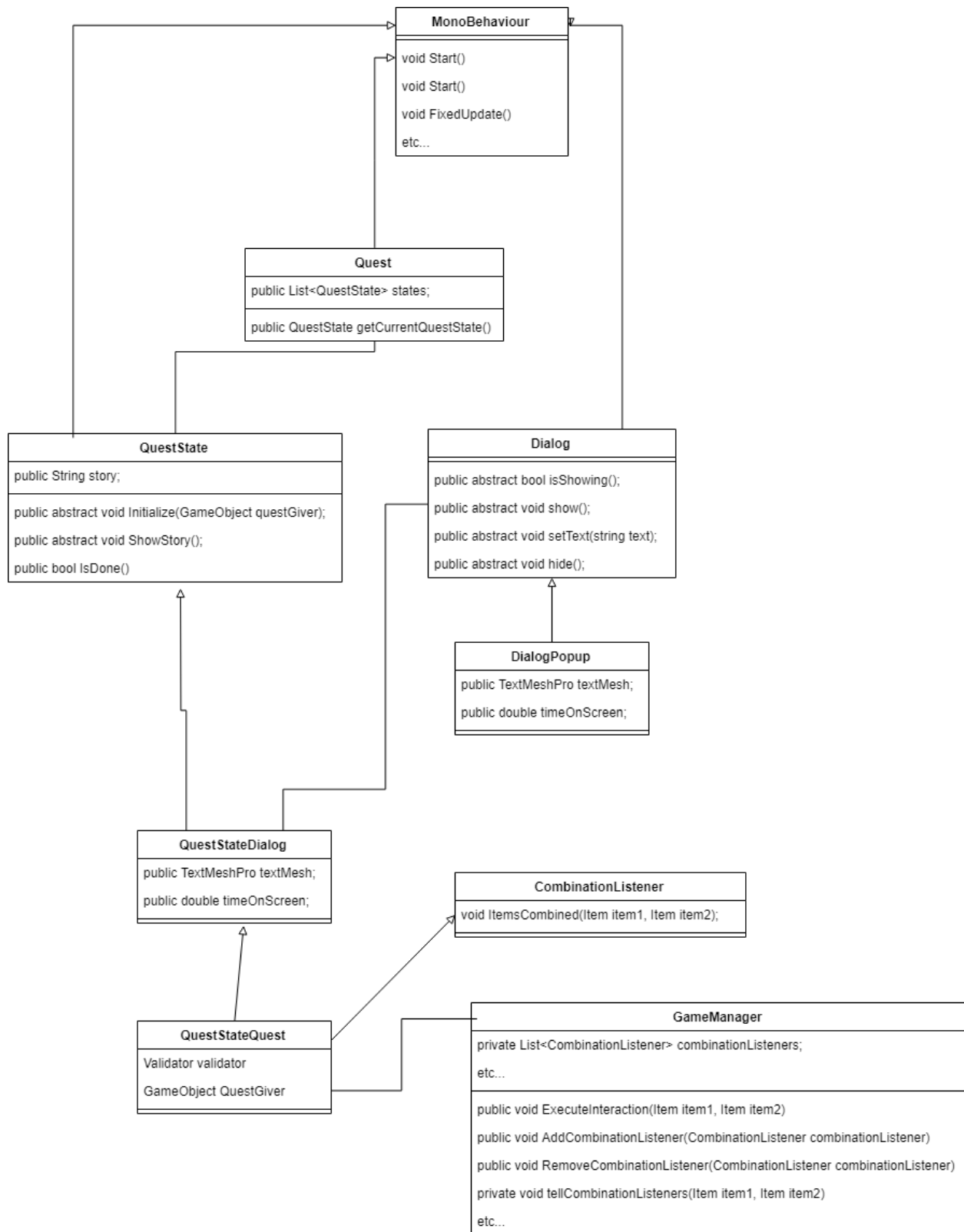
- ItemValidator uses the name inherited from ScriptableObject

4.1.5 Effect

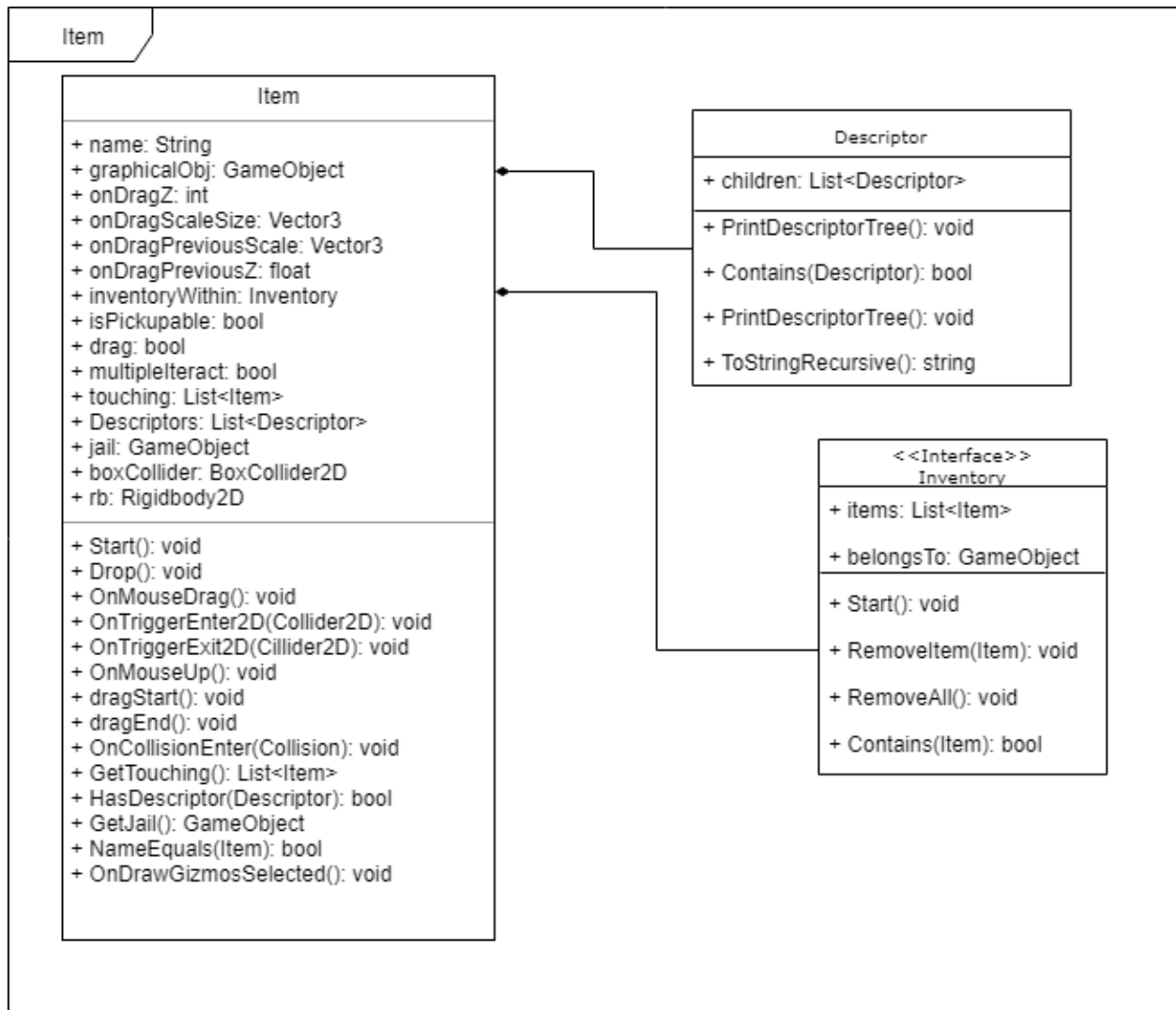


- Teleport uses the jail property from Item
- EffectRespawnable uses Respawnable's position

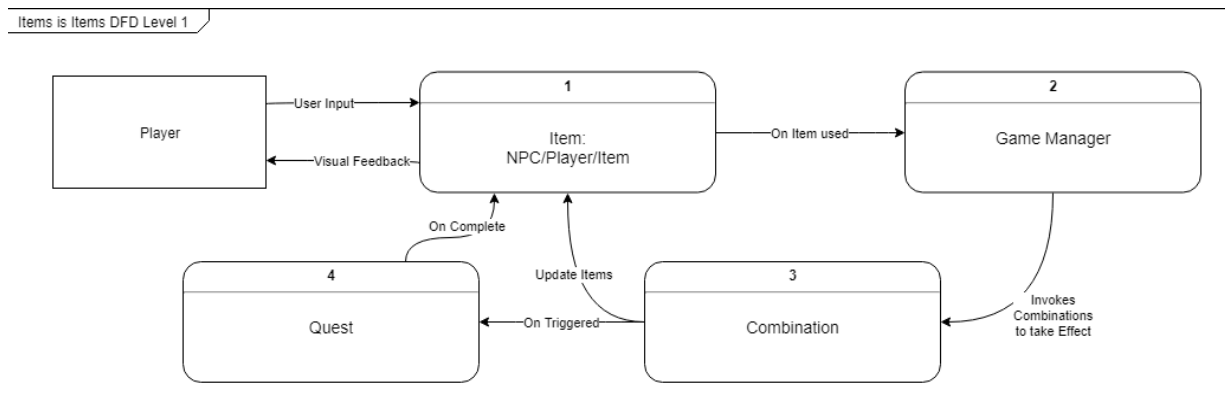
4.1.6 Quest:



4.1.7 Item:



4.2 Data Flow Diagram



5. RTM (Requirements Traceability Matrix)

Req. #	Requirement	Interface Design Component	Design Specification	Test Case ID
1	Menu Functionalities	2.2.1	3.3.1	1
2	Item Functionalities	2.2.2.3	3.3.2	4-5
3	Character Capabilities	2.2.2.1	3.3.3	2
4	NPC Capabilities	2.2.2.2	3.3.4	3