

Items Is Items

Software Test 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

Index

Index	2
Introduction	3
Purpose	3
Scope	3
Test Approach	3
References	Error! Bookmark not defined.
Test Plan	4
2.1 Features to be tested	4
2.2 Features not to be tested	4
2.3 Testing Tools and Environment	4
Test Cases	4
Requirements Traceability Matrix	Error! Bookmark not defined.
Responsibilities	9
Staffing and Training Needs	Error! Bookmark not defined.
Schedule	9

Introduction

Purpose

The purpose of this document is to describe the tests that will be used in the Items is Items game project.

Scope

Items is Items is a game that allows casual play. Our tests will keep our code working as we continually update it and ensures that the gameplay functions and user requirements actually work (things like player mobility, the collection of items, and the items uses).

Test Approach

The test approach in Unity is to create a scene containing the object that will be used to test. Perform an operation. Wait. And then check for the correct output.

Test Plan

2.1 Features to be tested

1. Menu
 - a. World Saving
 - b. World Loading
2. Movement
3. NPC
 - a. NPC moves to target (Items they are attracted to, hated, or away from items they are scared of)
 - b. NPC automatically uses items that they touch
4. Inventory
 - a. Player colliding with an item puts that item into the player's inventory
 - b. Items in the inventory follow the inventories owner aka the player
5. Item Interactions other Items
6. Quests

2.2 Features not to be tested

- Unity Loading
- Unity's Physics System
- Accessibility
- Response Times

2.3 Testing Tools and Environment

- Unity3D
 - Unity Test Runner

2.3.1 Unity Test Runner

Built into the code-base for Unity, Unity Test Runner allows you to test code while in edit or in play mode. After creating a test script it can be run and the results can be verified within Unity.

2.3.2 GCop

GCop is a code linter to improve C# code and can be installed as an extension in Visual Studio 2017. (Brief description of linter: edits code styling)

3. Test Cases

Test Case	Menu	Purpose	Input	Expected Output	Pass/Fail Criteria
-----------	------	---------	-------	-----------------	--------------------

ID					
1.a	Save World	Able to Save the world	World, Save function	A new world file	Expected Output Passes
1.b	Load World	Loading the world works	Given the url for a world file	The world is loaded	Expected Output Passes

Test Case ID	Movement	Purpose	Input	Expected Output	Pass/Fail Criteria
2.a	Player Movement	Player is able to move	Character, Target Position, Time	Character arrives at Target Position	Expected Output Passes

Test Case ID	NPC	Purpose	Input	Expected Output	Pass/Fail Criteria
3.a	NPC Wander Movement	NPC's are able to wander within a set area of their designated position	NPC, Time	Every tick, there is a random chance the NPC will select a distance to move to, and moves to it.	NPC moves randomly within a bounded area
3.b	NPC Purpose Movement	NPC's are drawn to item in their field of view	NPC, Desired Item, Time	If a desired item is in a NPC's field of view, the NPC will move towards that item over a period of time	NPC gradually moves towards item until it arrives at item position
3.c	NPC Return Movement	NPC's return to their home after wandering	NPC, Time	After timer runs out and there is no other item target, NPC will return to where it was	NPC moves towards its home base
3.d	NPC Picks Up Item	If an NPC walks over an item, it is added to their	NPC, Item	When item collision is detected with NPC, item is placed in inventory	Item is in NPC's inventory and is no longer visible to the world

		inventory			
3.e	NPC Smells Item	If an item is in smell range and not viewable, NPC rotates	NPC, Out of View Item	The NPC will rotate to look for item until it is within view	NPC rotates

Test Case ID	Inventory	Purpose	Input	Expected Output	Pass/Fail Criteria
4.a	Putting Item into Inventory	Character able to store item	Character steps on item	Item goes behind player	Expected Output Passes
4.b	Item follows Player	Item should follow the player	Character moves, Item in Inventory	Item should still be within a certain distance from player	Expected Output Passes
4.c	Selectable Inventory Item	Player able to use items in his inventory	Player clicks/taps item in inventory	Item should be selected	Expected Output Passes
4.d	Move item from Inventory to world	Player can select an item in their inventory and place it in the world	Player clicks item in inventory, and clicks item when cursor is on world	Item is removed from player's inventory. Item is now an object in world	Expected Output Passes

Test Case ID	Item	Purpose	Input	Expected Output	Pass/Fail Criteria
5.a	Use Item on another Item	Selected Item can be used on other items	Selected Item Flower Yellow Used and click on Item Flower Blue	Item Flower Yellow Gone, Item Flower Blue now Item Flower Purple	Expected Output Passes
5.b	Use Item on Player	Selected Item can be	Selected Item Flower Yellow	Player Color now Yellow	Expected Output Passes

		used on the player	Used and click on Player		
5.c	Use Item on NPC	Selected Item can be used on NPCs	Selected Item Flower Yellow Used and click on NPC Watchman	NPC Watchman Color now Yellow	Expected Output Passes
5.d	Use Item on Structure	Selected Item can be used on structures	Selected Item Flower Yellow Used and click on Mill	Item Flower Yellow now is Item Powder Yellow	Expected Output Passes

Requirements Traceability Matrix

Req. #	Requirement Description	Data Design Component	Interface Design Component	Test Case #
1	Menu Functionalities: Covers the functionalities each menu item (Main Menu, in-game Menu) will have.	2.2.1	3.3.1	1
2	Item Functionalities: Covers the manipulation and use of items.	2.2.2.3	3.3.2	4-5
3	Character Capabilities: Covers character movement and item use.	2.2.2.1	3.3.3	2
4	NPC Capabilities: Covers NPC movement	2.2.2.2	3.3.4	3

Responsibilities

The testers should ensure that the dependant functions have tests that will break if their interfaces are changed or if extraneous cases are used .

Schedule

Testing will occur on each feature as it is added. Currently no known schedule for testing as of the quarter is at an end.