



Shattered Seal: Echelon

Test Plan Document

Bryan Evans, bevans@my.fit.edu

Dan Baumann, dbaumann@my.fit.edu

Paul Graham, pgraham@my.fit.edu

Scott Leierer, sleierer@my.fit.edu

Stephen Garcia, sgarciam@my.fit.edu



Table of Contents

Table of Contents	i
Revision History	ii
1. Test Plan Description.....	1
1.1 Testing Objectives	1
1.2 Testing Approaches	1
2. Game Mechanics Tests	1
2.1 Artificial Intelligence	1
2.2 Models.....	1
2.3 Other Game Mechanics.....	2
3. Gameplay Tests	3
3.1 User Options	3
3.2 User Interactions with the Game Environment	3
3.3 Mission Objectives.....	4
3.4 Other Gameplay Tests.....	5
4. Interface Tests	6
4.1 Graphical User Interfaces.....	6
4.1.1 Main Menu.....	6
4.1.2 Pregame Interface	7
4.1.3 In-Game Interface	7
4.1.4 Postgame Interface.....	8
4.2 Hardware Interfaces	9
5. Audio Tests	10
6. Other Nonfunctional Tests.....	11
6.1 Performance Tests.....	11
6.2 Software Quality Attributes	11

1. Test Plan Description

1.1 Testing Objectives

Games, just like any other software, must be thoroughly tested in order to exceed expected user satisfaction.

1.2 Testing Approaches

Because every feature described in the requirements document is to be tested with the exception of constraints, specific approaches to testing will be implemented. Functional and non-functional requirements require a different approaches to testing. The functional requirements will be tested by using sample inputs and monitoring the outputs of the functions. Tests will use both correct and incorrect input in order to compare to the expected output. The non-functional requirements will use a combination of performance and usability tests to determine if the requirements have been adequately met.

The software itself will undergo thorough alpha and beta testing following the development stage. This testing will essentially be further usability testing of the overall product.

2. Game Mechanics Tests

2.1 Artificial Intelligence

Requirement	Name	Release History
3.1.0.0	Enemy AI	1.0 (<i>September 18, 2009</i>)
Test Description: Observe that the enemy units follow algorithms that make the units engage and interact with the user and his/her movements.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
3.1.0.1	Anti-trap Measures	1.0 (<i>September 18, 2009</i>)
Test Description: Observe the movement patterns of enemy units and make sure that no unit gets stuck in any location.		
Priority: 5	Dependencies: 3.1.0.0	Type: non-functional

2.2 Models

Requirement	Name	Release History
3.2.0.0	2D Models	1.0 (<i>September 18, 2009</i>)
Test Description: Observe that everything is being rendered. All models should be represented through sprites that are independent		

from the background.		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional
3.2.0.1	Lighting Effects	1.0 (<i>September 18, 2009</i>)
Test Description: Light ray intersection will be the source of testing. With an in-game object being designated as being a light source, tests can be performed based on expected light values on the surfaces of objects. This specifically focuses on objects casting shadows.		
Priority: 2	Dependencies: 3.2.0.0	Type: functional
3.2.0.2	Enemy and Player Models	1.0 (<i>September 18, 2009</i>)
Test Description: Observe that there are different models for the player and enemy characters.		
Priority: 4	Dependencies: 3.2.0.0	Type: non-functional
3.2.0.3	Scenery Models	1.0 (<i>September 18, 2009</i>)
Test Description: Observe that there are models for objects in the scenery. This includes the background as well as miscellaneous objects found throughout the setting.		
Priority: 2	Dependencies: 3.2.0.0	Type: non-functional

2.3 Other Game Mechanics

Requirement	Name	Release History
3.3.0.1	Basic Combat Physics	1.0 (<i>September 18, 2009</i>)
Test Description: The tests for this requirement will vary in number by the amount of different firing patterns that the development team decides to use. For example, if one gun fires like a shotgun with a simple spray that fires straightforward and another gun fires a Ghostbuster-like energy beam that whips around, then there will be a simple test for each one.		
Sample Input: Right analog stick is pressed to the left and then, in an arcing, clockwise motion, is moved to the upward position (this should simulate a player firing to the left and then directing their fire upwards).		
Expected Output: Depending on the type of weapon fire, the weapon should display a uniqueness in the firing as the direction changes.		
Priority: 5	Dependencies: <i>none</i>	Type: functional
3.3.0.2	Kill Statistics	1.0 (<i>September 18, 2009</i>)
Test Description: A kill should increase the kill count variable by one.		
Sample Input: (this is more of a void function and as such, wouldn't have an input)		
Expected Output: The kill count variable is increased in value by one integer.		

Priority: 2	Dependencies: <i>none</i>	Type: functional
-------------	---------------------------	------------------

3. Gameplay Tests

3.1 User Options

Requirement	Name	Release History
4.1.0.1	Multiple Usable Weapons	1.0 (<i>September 18, 2009</i>)
Test Description: Each weapon will be tested to see if the player character can equip and use the weapon.		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional
4.1.0.2	Multiple Usable Spells	1.0 (<i>September 18, 2009</i>)
Test Description: Each spell will be tested to see if the player character can select and cast the spell.		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional
4.1.0.3	Multiple Usable Tech Abilities	1.0 (<i>September 18, 2009</i>)
Test Description: Each tech ability will be tested to see if the player character can select an ability. Then, testing will involve the observation of these abilities altering the character traits as they are intended to.		
Priority: 4	Dependencies: <i>none</i>	Type: non-functional
4.1.0.4	Talent Trees	1.0 (<i>September 18, 2009</i>)
Test Description: Multiple game files will be used to create player characters that will each explore a different path down the talent trees. Each of these characters must be able to successfully make it to the leaf of their part of the tree. Also, each character must have separate, unique abilities in comparison to each other.		
Sample Input: User selects to spend a talent point towards a specific child of the current node in the talent tree that they are on.		
Expected Output: The talent abilities are altered to now reflect the current node in the talent tree.		
Priority: 4	Dependencies: <i>none</i>	Type: non-functional

3.2 User Interactions with the Game Environment

Requirement	Name	Release History
4.2.0.1	Movable Player Character	1.0 (<i>September 20, 2009</i>)

<p>Test Description: Movement input manipulates the player character's position in the game world.</p> <p>Sample Input: Left analog stick is pressed to the left.</p> <p>Expected Output: Player character moves to the left in comparison to the background scenery.</p>		
Priority: 3	Dependencies: <i>none</i>	Type: functional
4.2.0.2	Anti-Grinding Measures	1.0 (<i>September 19, 2009</i>)
<p>Test Description: Observe during gameplay testing that grinding is not required in order for the player character to be strong enough to move onto the next mission.</p>		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional
4.2.0.3	Hurt Enemies	1.0 (<i>September 19, 2009</i>)
<p>Test Description: Attack an enemy unit to make sure that it takes damage.</p> <p>Sample Input: 50 damage points</p> <p>Expected Output: take away 50 points from the unit's current total health.</p>		
Priority: 5	Dependencies: <i>none</i>	Type: functional
4.2.0.4	Move Through Enemy Dead Units	1.0 (<i>September 19, 2009</i>)
<p>Test Description: Kill an enemy unit and then attempt to move through the object representing the remains. The enemy unit will be located to the right of the player character and then killed.</p> <p>Sample Input: Left analog stick is pressed to the right.</p> <p>Expected Output: Player character moves right and passes through the model of the dead enemy.</p>		
Priority: 5	Dependencies: <i>none</i>	Type: functional
4.2.0.5	Boss Battles	1.0 (<i>September 20, 2009</i>)
<p>Test Description: Observe that at the end of each mission, there is a boss battle.</p>		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional

3.3 Mission Objectives

Requirement	Name	Release History
4.4.0.0	Mission Objectives	1.0 (<i>September 20, 2009</i>)
<p>Test Description: Observe that each mission requires the user to complete at least one objective in order to complete the mission.</p>		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional
4.4.0.1	Optional Objectives	1.0 (<i>September 20, 2009</i>)

Test Description: Observe that at least one mission provides an optional mission that is not required to be completed.		
Priority: 3	Dependencies: 4.4.0.0	Type: non-functional
4.4.0.2	Optional Objective Replay	1.0 (September 20, 2009)
<p>Test Description: During a replay of the mission, completed, optional objectives will already be marked as being completed (and uncompleted objectives still available for completion.</p> <p>Sample Input: The XML file holding the level completion information. Expected Output: Completed, optional objectives will become marked as already being completed.</p> <p>Sample Input: An optional objective is completed and the mission is finished. Expected Output: The XML file holding the level completion information is altered.</p>		
Priority: 1	Dependencies: 4.4.0.1	Type: non-functional
4.4.0.3	Objective Completion Notification	1.0 (September 20, 2009)
<p>Test Description: Mark that an objective is completed and make sure that there is an in-game response.</p> <p>Sample Input: Mark that an objective is complete. Expected Output: In-game message that the objective was completed.</p>		
Priority: 3	Dependencies: 4.4.0.0	Type: functional

3.4 Other Gameplay Tests

Requirement	Name	Release History
4.4.0.1	No Quick Time Events	1.0 (September 20, 2009)
Test Description: Observe that there are no quick time events during any portion of the game.		
Priority: 1	Dependencies: <i>none</i>	Type: Non-functional
4.4.0.2	Changing Mana	1.0 (September 20, 2009)
<p>Test Description: Cast a spell to see if the mana value decreases and that that the visual display matches the mana's variable value.</p> <p>Sample Input: Cast a spell with a cost of 30 mana points. Expected Output: Mana variable is decreased by 30 points. Also, mana bar in in-game UI is reduced accordingly.</p>		
Priority: 5	Dependencies: <i>none</i>	Type: Functional
4.4.0.3	Changing Health	1.0 (September 20, 2009)

Test Description: Allow an enemy unit to attack the player character.		
Sample Input: 15 points of damage. Expected Output: Health variable is decreased by 15 points. Also, health bar in in-game UI is reduced accordingly.		
Priority: 5	Dependencies: <i>none</i>	Type: Functional
4.4.0.4	Multiple Save Files	1.0 (<i>September 20, 2009</i>)
Test Description: Make multiple save files and make sure that each loads correctly.		
Sample Input: XML file filled with information detailing the specifics of the user's gameplay. Expected Output: A new save file.		
Priority: 5	Dependencies: <i>none</i>	Type: Functional

4. Interface Tests

4.1 Graphical User Interfaces

4.1.1 Main Menu

Requirement	Name	Release History
5.1.1.0	Main Menu	1.0 (<i>September 16, 2009</i>)
Test Description: Observe that upon starting the software, the main menu is displayed.		
Sample Input: Software is initialized. Expected Output: The main menu is displayed.		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional
5.1.1.1	Menu with Logo	1.0 (<i>September 16, 2009</i>)
Test Description: Observe that the logo appears on the main menu.		
Priority: 1	Dependencies: 5.1.1.0	Type: non-functional
5.1.1.2	Load Feature on Menu	1.0 (<i>September 16, 2009</i>)
Test Description: Make sure that saved files are displayed. Also, save files must all be selectable and run when selected.		
Sample Input: Load Game is selected. Expected Output: Available save files are displayed.		
Sample Input: Save file is selected Expected Output: The game begins at the point indicated by the save file.		
Priority: 3	Dependencies: 5.1.1.0	Type: functional

5.1.1.3	Exit Game	1.0 (September 18, 2009)
<p>Test Description: Software is terminated by selecting the "Exit Game" option.</p> <p>Sample Input: Exit Game is selected.</p> <p>Expected Output: Game terminates and returns to the Xbox 360 dashboard.</p>		
Priority: 5	Dependencies: 5.1.1.0	Type: functional

4.1.2 Pregame Interface

Requirement	Name	Release History
5.1.2.1	Mission Briefing	1.0 (September 18, 2009)
<p>Test Description: Observe that there is a mission briefing before each mission.</p>		
Priority: 3	Dependencies: 5.1.2 4.4.0.0	Type: non-functional
5.1.2.2	Mission Objective Screen	1.0 (September 18, 2009)
<p>Test Description: Observe that there is a mission objective screen following the mission briefing.</p>		
Priority: 3	Dependencies: 5.1.2 4.4.0.0	Type: non-functional
5.1.2.3	Choose Loadout	1.0 (September 18, 2009)
<p>Test Description: Select various combinations of spell and weapons to make sure that they are loaded in the user's loadout.</p> <p>Sample Input: Weapon A</p> <p>Expected Output: Weapon A is in the player character's loadout during the mission.</p>		
Priority: 3	Dependencies: 5.1.2 4.1.0.1 4.1.0.2	Type: functional

4.1.3 In-Game Interface

Requirement	Name	Release History
5.1.3.0	In-Game UI	1.0 (September 16, 2009)
<p>Test Description: Observe that a user interface is displayed during the gameplay of the mission.</p>		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional
5.1.3.1	Health Bar	1.0 (September 16, 2009)
<p>Test Description: Observe that a graphic representation of the user's</p>		

<p>in-game health attribute is displayed on the interface.</p> <p>Sample Input: 45 health points from the variable holding the health attribute values.</p> <p>Expected Output: A bar that is filled at 45 divided by the maximum amount of health points the player character's has at full health.</p>		
Priority: 4	Dependencies: 5.1.3.0	Type: functional
5.1.3.2	Mana Bar	1.0 (September 16, 2009)
<p>Test Description: Observe that a graphic representation of the user's in-game mana attribute is displayed on the interface.</p> <p>Sample Input: 30 mana points from the variable holding the mana attribute values.</p> <p>Expected Output: A bar that is filled at 30 divided by the maximum amount of mana points the player character's has at full mana.</p>		
Priority: 4	Dependencies: 5.1.3.0	Type: functional
5.1.3.3	Character Portrait	1.0 (September 16, 2009)
<p>Test Description: Observe that there is a character portrait display in the in-game user interface.</p>		
Priority: 2	Dependencies: 5.1.3.0	Type: non-functional
5.1.3.4	Display Ability Cooldown	1.0 (September 16, 2009)
<p>Test Description: Use a cooldown enforced ability and make sure that the cooldown rules are enforced.</p> <p>Sample Input: the use of a cooldown enforced ability</p> <p>Expected Output: countdown timer displayed on user interface. Also, the same ability is locked (not functioning) during the cooldown period.</p>		
Priority: 2	Dependencies: 5.1.3.0 5.1.3.3	Type: functional
5.1.3.5	Control Scheme Display	1.0 (September 21, 2009)
<p>Test Description: Check to see if the display is accessible during gameplay.</p> <p>Sample Input: Back button is pressed and held</p> <p>Expected Output: Window with the control scheme is displayed in the user interface.</p>		
Priority: 2	Dependencies: none	Type: functional

4.1.4 Postgame Interface

Requirement	Name	Release History
5.1.4.1	Mission statistics	1.0 (September 18, 2009)
<p>Test Description: A test run of a mission is performed. During this test run, a check list is used to make sure that each part of the mission statistics has had a chance to receive input and change. Following the</p>		

<p>mission, the statistics can me compared to their expected result.</p> <p>Sample Input: Gameplay during a run through a mission. Expected Output: Alteration in the total play time and, if any kills occurred during gameplay, an increase in the total kill count.</p>		
Priority: 3	Dependencies: 5.1.4 3.3.0.2	Type: functional
5.1.4.2	Option to Quit	1.0 (September 18, 2009)
<p>Test Description: The tester will select the option to quit and make sure that it returns the user back to the main menu.</p> <p>Sample Input: Quit Game button selected. Expected Result: the Main Menu interface is displayed.</p>		
Priority: 5	Dependencies: 5.1.4	Type: functional
5.1.4.3	Option to Retry	1.0 (September 18, 2009)
<p>Test Description: The tester will select the option to retry and make sure that it returns the user to the mission briefing for the mission he/she had just completed.</p> <p>Sample Input: Retry button selected. Expected Result: the previous Mission Briefing interface is displayed. Also, the mission statistics are wiped and any objectives completed are reset.</p>		
Priority: 4	Dependencies: 5.1.4	Type: functional
5.1.4.4	Ability to Save Game	1.0 (September 18, 2009)
<p>Test Description: The tester will select the option to save and make sure that it creates an appropriate save file in the selected slot.</p> <p>Sample Input: Save button is selected. Expected Result: A list of available save slots is displayed.</p> <p>Sample Input: Save slot is selected. Expected Result: the XML file that is the save file is sent to be stored on the Xbox 360's hard drive.</p>		
Priority: 4	Dependencies: 5.1.4	Type: functional
5.1.4.5	Level Up Interface	1.0 (September 18, 2009)
<p>Test Description: Observe that the level up interface appears after the mission summary.</p>		
Priority: 5	Dependencies: 5.1.4	Type: non-functional

4.2 Hardware Interfaces

Requirement	Name	Release History
5.2.0.1	Controller Scheme	1.0 (September 16, 2009)

<p>Test Description: A series of usability tests will be performed on the controller and the controller configuration in order to make sure that the settings feel natural.</p>		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional
5.2.0.2	Left Analog Stick for Movement	1.0 (<i>September 17, 2009</i>)
<p>Test Description: Moving the left analog stick in order to make sure that it moves the player character.</p> <p>Sample Input: Left analog stick is pushed in a clockwise rotation. Expected Result: The player character will move in a clockwise circle.</p>		
Priority: 5	Dependencies: 5.2.0.1	Type: functional
5.2.0.3	Right Analog Stick for Aiming and Shooting	1.0 (<i>September 17, 2009</i>)
<p>Test Description: Moving the right analog stick in order to make sure that it directs the fire.</p> <p>Sample Input: Right analog stick is pushed in a clockwise rotation. Expected Result: The player character will fire (either a weapon or a spell) in a clockwise rotation around the player character.</p>		
Priority: 5	Dependencies: 5.2.0.1	Type: functional

5. Audio Tests

Requirement	Name	Release History
6.0.0.1	Sound Effects	1.0 (<i>September 20, 2009</i>)
<p>Test Description: Perform an action in-game that is guaranteed to be tied to a sound effect. An action must be found for each and every sound effect.</p> <p>Sample Input: Complete a mission objective. Expected Result: A notification sound should play through the speakers.</p>		
Priority: 3	Dependencies: <i>none</i>	Type: functional
6.0.0.2	Background Music	1.0 (<i>September 20, 2009</i>)
<p>Test Description: Play each mission and run through the interfaces to make sure that music is playing for all of them.</p> <p>Sample Input: A mission is completed. Expected Result: Music for the mission statistics interface will play through the speakers.</p>		
Priority: 3	Dependencies: <i>none</i>	Type: functional

6. Other Nonfunctional Tests

6.1 Performance Tests

Requirement	Name	Release History
7.1.0.1	100 Millisecond Response Time	1.0 (<i>September 16, 2009</i>)
Test Description: Run performance tests to make sure that the average response time is at least under 100 milliseconds. Tests will include high stress as well.		
Priority: 3	Dependencies: <i>none</i>	Type: non-functional

6.2 Software Quality Attributes

Requirement	Name	Release History
7.2.0.1	High Reliability	1.0 (<i>September 16, 2009</i>)
Test Description: Observe that the software shows a high level of reliability.		
Priority: 5	Dependencies: <i>none</i>	Type: non-functional