

**2/22/2006**

**Team #7: Pez**

**Project: Empty Clip**

**Members: Alan Witkowski, Steve Huff, Thos Swallow, Travis Cooper**

**Document: VVP**

## **1. Introduction and overview**

### **1.1 Purpose of this Document**

The purpose of this document is to outline the overall quality assurance procedures, including testing, and validating new features.

### **1.2 Scope of the Development Product**

The project is a top down RPG that is designed to allow 1-4 players to complete levels and tasks, and to kill monsters and each other. The user will use keyboard and mouse to control the character and will use the UI and the game environment to interact with the game.

### **1.3 Definitions, Acronyms, and Abbreviations**

2D – 2 dimensional.

LAN – Local Area Network.

MB – Mega Byte

RPG – Role Playing Game

UI – User Interface

### **1.4 References**

[SDS](#)

[SRS](#)

### **1.5 Overview of Document**

Section 2 will talk about reviews, walk-throughs, inspections, and audits.

Section 3 will talk about component test plans and procedures.

Section 4 will talk about system test plans and procedures.

Section 5 will talk about how we locate and correct defects.

Section 6 will talk about the relationships between the SRS and SDS.

Section 7 will talk about the testing requirements for each component based on the traceability matrix.

Section 8 will talk about the criteria for an acceptable release.

Section 9 will talk about any additional information we need for this project.

## **2. Reviews, walkthroughs, inspections, and audits**

Every week we will review the previous week's work according to our schedule and goals. We will conduct a walkthrough before each release to see what needs to be fixed. The team leader will conduct inspections to make sure integration stays intact between team members.

We have been following our original plan for reviewing progress made by each team member, and have been conducting technical reviews of the code. We have also followed the schedule for reviews, which is every Monday. We have a walkthrough scheduled for our first release, which will take place on March 1<sup>st</sup>.

### **3. Component test plans and procedures**

#### **3.1 Component test strategy overview**

Our game will consist of four main modules. These components are: graphics, menu/UI, audio, and networking. We will test these components individually and as a group to make sure that integration is working properly. As a new feature is added to a specific module we will test that new feature alone, and in the setting of the entire game.

#### **3.2 Graphics**

The graphics will be tested in the map editor and in the game play. The graphics will need to be visually appealing, believable and understandable. As a new feature is added we can add it to the map via the map editor and during game play we can test to see if the new feature is working properly. We will also add additional code, for testing purposes, to help us in testing this new feature. Components from our test matrix in this category are as follows: user can use a weapon, monster can be attacked and killed, items can be picked up/put down and used, the player can maneuver around the map (including barriers), user can draw blocks, and delete blocks through the map editor.

#### **3.3 Menu/UI**

The menu and UI will be tested in the game play and in the menu previous to the game. The menu and UI will need to have properly working buttons that relay to the user the current state of the game. In order to test the menu and UI we will need to make sure that each button is working in the expected manner and that the information is being relayed from the menu to the game and visa versa without loss of data. The user must be able to interact with the UI and the game. The components from our test matrix in this category are as follows: the user can create a player, can change keyboard configurations, can save a game state, can load a game state, can store weapons, items, and ammo, the user can gain points for experience, and can complete tasks to advance in levels.

#### **3.4 Audio**

The audio will be tested in the game play and in the main menu. The audio will help the user to know what is happening during game play. For example, when the monsters are coming, when you are out of ammo, etc. The audio will also provide background music to keep the user addicted. The components from the matrix are as follows: audio should be heard for appropriate actions.

#### **3.5 Networking**

The networking will be tested in multi-player mode of the game play. Multi-player mode includes 2-4 players across a LAN. The networking must be fast and accurate so that each player will see the same game state and it will happen in real time so the user won't get frustrated by bad information. The components from the matrix are as follows: multiple players should be able to interact together to kill monsters, each other, and beat game levels.

## 4. System test plans and procedures

### 4.1 System test strategy overview

<i>Component</i>	<i>Details</i>
Testing process	The testing procedure described in this plan, given in terms of phases and activities.
Requirements traceability	Relate the testing to the requirements given in the SRS
Components Tested	The items being tested; identify them as precisely as possible.
Testing schedule and resources	The overall testing schedule and resource allocation.
Test recording procedures	How the results of the tests will be systematically recorded. It must be possible for an outside agency to inspect the test plan, the test cases, and the test results to determine that the testing process has been carried out correctly.
Hardware and software requirements	The software tools that are needed for running these tests as well as any hardware that will be needed.
Constraints	Any constraints that will affect the testing process.

### 4.2 Gameplay

Test case group identification	This is the main component of our test plan.
Features to be tested	Need to test the following components and their interaction: user interaction with the interface, graphics, audio, and networking.
Testing approach	<p>We will gather a group of beta testers to play through various levels and report issues with graphical glitches, gameplay bugs, and networking errors.</p> <p>We will look through the reports and assign priority to each gameplay issue. We will then go through the list and fix the bugs.</p> <p>We feel that by using an outside group, there is a higher probability of finding rare defects. In addition, using more people increases the likelihood of testing all of the components of the game.</p>
Pass/Fail criteria	We will evaluate the game by counting the number of severe defects by the release stage. If there are more than five serious defects, then the project fails.
Individual test cases	<p>Testing levels</p> <ul style="list-style-type: none"><li>• <i>Input:</i> Keyboard and mouse</li><li>• <i>Output:</i> Graphics and audio correctly identify the game state.</li><li>• <i>Environment:</i> Need to have a computer with 32 MB video card. Tester should not be a developer</li><li>• <i>Special Procedures:</i> Play through the level and test every aspect of it</li></ul>

Test case group identification

This is the main component of our test plan.

- *Precedence and dependencies:* The levels should be tested in the order that they would normally be played

### 4.3 Balance

Test case group identification

This comes after our gameplay system test

Features to be tested

Need to test the difficulty of missions, the flow of the game, power of weapons, ease of leveling, and replay value.

Testing approach

We will gather a group of beta testers to play through various levels and report balance issues.

We will look through the reports and assign priority to each balance issue. We will then go through the list and consider improving the balance.

We feel that by using mixed group of gamers, the game will end up being enjoyable by a wider audience.

Pass/Fail criteria

We will evaluate the game by counting the number of severe balance issues by the release stage. If there are more than five serious issues, then the project fails.

Individual test cases

Testing levels

- *Input:* Keyboard and mouse
- *Output:* Frustration and happiness levels on a scale from 1-10
- *Environment:* Alone in a dark room with no interruptions.
- *Special Procedures:* Start a new character, play through the game as they would normally.
- *Precedence and dependencies:* Test the difficulty of the levels first, then test replay value.

## 5. Defect tracking plans

We have a table, included on the website, that will help us to track our defects. As a defect is detected the person who finds it will fill out a new defect table and will let the other members know of the defect, through e-mail. At this point the appropriate person will try to recreate the defect and take measures to fix it. If there is additional help needed communication through our Monday meetings and e-mail can be used. We will try to keep defect turnaround time in under 6 working hours. The severity will be measured on a 1-4 scale 1 being the worst, as described on our defect page.

## 6. Traceability from SRS to SDS

	Env	UI	Graphics	Character	Monsters	Objects	Textures	Physics / CD	Menu
Game function	x	x	x	x	x	x	x		
Game control	x	x		x				x	x
User interface		x	x						x
Networking	x	x		x	x	x		x	

## 7. Test-requirements cross-reference matrix

This is defined in the traceability matrix found [here](#).

## 8. Acceptance test and preparation for delivery

### 8.1 Procedure by which the software product will be acceptance tested

The person adding the new feature will be required to test each aspect of the feature until it interacts with the rest of the game without any known bugs before committing the changes. If any further defects are found the bug will be added to the defect table and this person will be asked to fix the defect.

### 8.2 Specific acceptance criteria

- New feature doesn't slow down game play
- Game doesn't have any known crashes due to the new feature
- New feature must interact with the other appropriate components in a realistic life like way.
- Must not interfere with any previous components.

### 8.3 Scenario by which the software product will be installed

We will have two install files available from our website. One install file will be for users without the .NET framework on their systems and the other for people that have previously installed the .NET framework. After downloading these install files the user will simple run the file and follow the on screen instructions to properly install the game on their system. The system must be running a windows OS, with a 3D-accelerated graphics card.

## 9. Additional information

None at this time.