|  |
| --- |
| Requirement driven testing |
| Test Plan for |
| Empires of the IV |
|  |
| **Requirement Driven Testing** |
| **11/5/2014** |

|  |
| --- |
| Andrew Godfroy |

# Approval

|  |  |  |  |
| --- | --- | --- | --- |
| AUTHOR | | | |
| Andrew Godfroy | Project Lead |  | |
| APPROVED BY | | | |
| Andrew Godfroy | Project Lead |  |  |
| ACKNOWLEDGMENT | | | |
| Acknowledged | | | |

# Introduction

This test plan document describes the scope, approach, resources and schedule of intended testing activities to be undertaken for Empires of the IV. This document should be read in conjunction with the Test Strategy.

## Purpose

This document provides the following guidance:

* Testing Scope;
* Entry and exit criteria for each test level
* A description of resources and tools to be used to conduct testing;
* An overview of test schedules per development cycle;
* An overview of the types of testing that is to be conducted;
* Defect management work flow.

## Project Overview

The Empires of the IV game is currently in the stage of Engine Development, as such this report will pertain to aspects involved in the development of the Anarian Game Engine.

# Testing objectives

Testing objectives around the Anarian Game Engine are focused around ensuring the Rendering and generic shader units are operating as required as it goes through Scene Nodes and renders models in correct succession to their individual sharers.

## Features to be tested

Rendering of Scene Nodes in correct succession

GameObject movement, rotation and scaling

Camera movement and rotations

Material and Mesh manipulations

Lighting

## Features Not to be tested and constraints

Object Picking

Mesh Animations

Testing Approach

## Static Testing

Static testing is testing of a component or specifications without execution of that software. This is usually done as soon as acceptance criteria or business requirements are ready for review before code implementation such as conflicting rules, invalid data types, redundant process just to name a few.

## Component Testing

Component level testing focuses on the functionality of each component being developed. This is crucial where different components are being developed before they are integrated together as one system.

### Entry Criteria

Component Testing may commence when the following criteria have been satisfied:

1. Component has been fully updated as per sprint requirements
2. All code have been unit tested and passed.
3. Test environment including software have been setup and configured correctly.

### Suspension Criteria

Component testing will be suspended under the following condition:

1. Critical error(s) found preventing test completion.
2. Change of requirements.
3. Change of environment components or technology including different version.

### Resumption Criteria

Component testing will resume when the following criteria are met:

1. All issues in suspension criteria have been resolved or mitigated
2. New software build has been redeployed or;
3. New build with fixed Critical and Medium severity defects has been deployed into Test.

### Exit Criteria (Test Completeness)

Component testing can be considered complete when the following conditions have been met:

1. All High and Medium priority requirements have been tested without Critical or Medium severity defects.
2. Program runs as it should as per project milestone goals

## System Testing

The purpose of the system testing is to validate that the complete and integrated system complies with functional requirements and business requirements.

### Entry Criteria

System testing may commence when the following criteria have been satisfied:

1. Component Testing has been completed.
2. No change to business requirements and test cases are up to date.
3. Scenario based test cases have reviewed by project management

### Suspension Criteria

System Testing will be suspended under the following condition:

1. Critical error(s) found affecting functionality of the whole system.
2. Change of project requirements
3. Critical error(s) found affecting main components of the system.

### Resumption Criteria

System Testing will resume when the following criteria have been satisfied:

1. All issues in suspension criteria have been resolved or mitigated
2. New build with fixed Critical and Medium severity defects has been deployed into Test.

### Exit Criteria (Test Completeness)

System Testing will be considered complete when the following conditions have been met:

1. All High and Medium priority requirements have been tested without Critical or Medium severity defects.
2. Project owner(s) and/or Project user(s) have been notified with any remaining defects and understand the risks or limitations of current release.
3. All defects found during testing have been recorded in defect management tool.

## User Acceptance Testing (UAT)

At the end of the sprint period, the Project will be shown off to supervisors for feedback and approval. During this period, notes will be taken and implemented during the next sprint.

## Test Environment Control

### Summary

I will be preforming all tests and procedures as needed and required by the project status.

### Release versioning

I operate my versioning based off of the M.M.B.R (Major.Minor.Build.Revision) versioning format. In general I will increment the Minor every week after each sprint cycle while updating the Build and Revisions as errors get resolved and submitted. At the end of the Sprint, Build and Revision is zeroed out to reset for the next sprint.

|  |  |
| --- | --- |
| **Version #** | **Description of version** |
| 0.1.0.0 | Basic Rendering Implemented |
| 0.15.0.0 | Rendering has been split up into components to allow a more object oriented approach to the Game Engine |
| 0.2.0.0 | .OBJ Models are able to be loaded and displayed on screen. Default Shaders have been written to allow for a generic shader structure when using the engine |
| 0.3.0.0 | Generic Data structures created to prepare for next step Picking has begun to be implemented, but is incomplete |
| 0.4.0.0 | Refactored Data structures to prepare for future steps. Added additional predefined Colors and Modified Camera Code to be more customizable.  An ICollider and Ray has been added to the program to allow for the future Picking implementation |
| 0.5.0.0 | Directional, Spotlight and Point Lighting has been implemented with Shader Code updated to support the new Buffers.  MD5Mesh loading has been fully implemented and a model can be drawn on screen from the MD5 File Format |

**Table**  **- Version numbering example**

## Testing Tool

CPPTest

# Appendix

http://cpptest.sourceforge.net/