

TEAM NAME

VIDEO GAME NAME TEST PLAN

Date: October 20, 2015

Prepared by: QA Tester name here

Table of Contents

INTRODUCTION	3
1.1 OBJECTIVES	3
1.2 TEAM MEMBERS	3
2 SCOPE.....	ERROR! BOOKMARK NOT DEFINED.
3 ASSUMPTIONS / RISKS	3
3.1 ASSUMPTIONS	3
3.2 RISKS.....	3
4 TEST APPROACH	4
4.1 TEST AUTOMATION.....	ERROR! BOOKMARK NOT DEFINED.
5 TEST ENVIRONMENT	4
6 MILESTONES / DELIVERABLES.....	5
6.1 TEST SCHEDULE	5
6.2 DELIVERABLES.....	5

Introduction

The test plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

1.1 Objectives

Put your goals and objectives from your OneNote page charter here. Delete this line when you do.

1.2 Team Members

Team Member Name	Role
name	Programmer
name	Project Manager
name	Tester/DBA
name	Assets

Assumptions / Risks

1.3 Assumptions

This section lists assumptions that are made specific to this project.

1. Delivery of the product is in format that the test team can check it into CVS.

1.4 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project.

Fill in risks on the chart and how you would mitigate those risks. Examples are included but you must do in your own words. Come up with at least one- 3 or more is preferred.

#	Risk	Mitigation Plan
1	Scope Increases – as testers and developers become more familiar with the game and game engine, they will want more functionality	Each iteration and its functionality will be closely monitored. Priorities will be set and discussed by the team.
3	Weekly delivery may not be possible because we will not have class time each week and may not be able to meet in person each week after school.	We will set aside certain days of the month to stay on track and deliver when possible or deliver through OneDrive from home.

2 Test Approach

Write in full sentences what you are using and why. There are a few parts to this.

Describe your different test types (Unit, Integration, and Functional are required- see below if you do not know what those mean).

Who is testing? For each type above list who will be doing the testing and what they will be doing.

Whole team approach- all members will test while QA does the bulk of testing and documentation of testing.

QA personnel only- QA will do the testing throughout the build process and record documentation on that testing.

Whole team and customer testing (Beta game perhaps?)- All members test including QA and having an Alpha or Beta version that you have customers test as well.

Defect management cycle:

How will your team log bugs? On the test case document at a minimum but they should have a separate log in OneNote or somewhere else.

Change management request:

How will the QA tester let the programmer know there is a change the needs made- will there be a submission form, on OneDrive, or something else?

You may not be familiar with some of the vocabulary so here's descriptions:

- **Unit tests-** Tests the parts of a game first. The goal of unit testing is to isolate each part of the game and show that the individual parts are correct. Examples: You will test player movements as a unit test, then you may test enemy AI, then you may test rewards, etc.
- **Integration tests-** Test an entire subsystem and ensure that a set of components play nicely together. So you will test all of the obj_player code- movements, collisions, rewards, etc as an integration test. Then you may test obj_player and obj_enemy to make sure all their code works together okay.
- **Functional tests-** Verify end-to-end scenarios that your users will engage in. This is where you actually play the game to test. You should do multiple scenarios like dying at different times, achieving different awards, checking multiple score intervals, etc.

3 Test Environment

Where will you test? In class, at home, at the library, a combination, etc. Put in full sentence.

4 Milestones / Deliverables

4.1 Test Schedule

FILL IN THESE DATES ACCORDING TO YOUR PLAN. DO NOT PUT FINISH UNTIL IT IS FINISHED. COMMENTS ARE ONLY AFTER AS WELL.

The initial test schedule follows:

Task Name	Start	Finish	Comments
Test Planning	09/20/2016		
Create test case document	09/23/2016		
First deploy to QA test environment- Unit tests			
Integration testing			
Functional testing			
Release to Production			

4.2 Deliverables

You can adjust these dates.

Deliverable	For	Date / Milestone
Test Plan	QA	September 2016
Test Cases	QA	September 2016
Test Results	Project Manager and QA	April 2017
Test Status report	QA	Throughout Project