



---

## **Team 12**

Ibrahim Mahmoud – Section 1 BN. 1

Ahmad Khaled – Section 1 BN. 3

Ahmad Essam – Section 1 BN. 7

Omnia Zakaria – Section 1 BN. 15

# **Geometry Adventure**

## **Software Test Plan (STP)**

**Version: 1.0**

**Configuration Identifier: STP-1.0**

## Revision History

Sl. No.	Prepared/ Modified by	E-mail	Version	Date	Approved by	Descriptions/ Remarks
1.	Ahmed Essam	theartful.ae@gmail.com	1.0	April 8, 2018	TBD	

Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

### **Distribution list**

<b>Name</b>	<b>E-mail</b>	<b>Notes</b>
Engineer Ali	alielseddeek@gmail.com	Review of test plan needed.

Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

## Table of Contents

1.	Overview	
1.1	Project Overview	5
1.2	Test Objectives	5
1.3	Test Scope	5
1.4	Test Approach	5
2.	Test Methodology	
2.1	Testing Environment	5
2.2	Functional Testing	5
2.3	Non-functional Testing	8
	2.3.1 Performance Testing	8
3.	References	9
4.	Glossary	9

Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

# 1. Overview

## 1.1 Project Overview

Geometry adventures is a cross-platform top-down action-shooter game in which the player fights his way through different levels.

## 1.2 Test Objectives

The purpose of the test plan is to make sure that the different aspects and functionalities of the game work according to the specifications. We shall not test all of the user scenarios and features at this stage as the project is still under development. But instead, we only test the implemented features. This shall serve as a metric and a guide to make sure that we are on the right path.

## 1.3 Test Scope

We test the implemented functional and non-functional requirements, which, broadly speaking, are:

- 1 – A main menu from which the user can start the game.
- 2 – Basic movement mechanics
- 3 – Basic enemy mechanics (patrolling, chasing the player).
- 4 – Specified non-functional requirements.

We also test the basic framework we developed for the game.

## 1.4 Test Approach

Our tests fall under three categories:

- 1 – Unit tests, which are designed to test a specific block of code. Most of the code cannot be covered by unit tests given the subjective nature of games.
- 2 – Functionality testing, which are designed to test the features of the game. This is done by playing the game and reporting bugs.
- 3 – Non-functional tests for constraints and performance.

# 2. Test Methodology

## 2.1 Testing Environment

Unit tests are done using the JUnit framework. We tested the ECS framework that we developed using unit tests, however, we are not going to expand on them in this document. Functional tests are done by playing the game and reporting on the behaviour.

## 2.2 Functional Testing

TC_ID	1	Test Case Name	User can move the player
<b>Purpose:</b>	Ensure that the user can control the movement of the player by dragging on the left side of the screen		
<b>Prerequisite:</b>	The player started a game level.		
<b>Priority:</b>	<b>High Priority</b>		
<b>Steps:</b>	1 – Start the game 2 – Choose a level. 3 – Move the player by dragging on the left side of the screen.		
<b>Expected</b>	The player should move with a speed proportional to the distance dragged.		

Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

<b>Results:</b>	
<b>Actual Results:</b>	The player did move with a speed proportional to the distance dragged.
<b>State:</b>	Pass.
<b>Issue #:</b>	N/A
<b>Execution Date:</b>	8.4/2018
<b>Build Version:</b>	Git.
<b>Notes:</b>	Movement system might be better if the player moved with an acceleration proportional to the distance dragged, and if a cap is put on the maximum speed.
<b>Tester Name:</b>	<b>Ahmed Essam</b>

<b>TC_ID ■ 2</b>	<b>Test Case Name ■ Camera and orientation test</b>
<b>Purpose:</b>	Ensure that the camera follows the player appropriately by making the player at the center of the screen while in landscape mode.
<b>Prerequisite:</b>	The player started a game level.
<b>Priority:</b>	<b>High Priority</b>
<b>Steps:</b>	1 – Start the game 2 – Choose a level. 3 – Move the player by dragging on the left side of the screen.
<b>Expected Results:</b>	The camera should follow the player.
<b>Actual Results:</b>	The camera did follow the player.
<b>State:</b>	Pass.
<b>Issue #:</b>	N/A
<b>Execution Date:</b>	8.4/2018
<b>Build Version:</b>	Git.
<b>Notes:</b>	The camera system is still basic. It might be better if the camera followed the player but with the position of the player not at the center. Instead, if the player is moving to the right for example, then he would be a little left to the screen, so as to show more of his path.
<b>Tester Name:</b>	<b>Ahmed Essam</b>

<b>TC_ID ■ 3</b>	<b>Test Case Name ■ User can attack using weapons</b>
<b>Purpose:</b>	Ensure that the user can using weapons by dragging on the right side of the screen
<b>Prerequisite:</b>	The player started a game level.
<b>Priority:</b>	<b>High Priority</b>
<b>Steps:</b>	1 – Start the game 2 – Choose a level.

Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

	3 – Attack by dragging on the right side of the screen.
<b>Expected Results:</b>	The weapon should initiate its specialized attack damaging any enemy in the path.
<b>Actual Results:</b>	The weapon did initiate its specialized attack, without affecting enemies.
<b>State:</b>	Fail.
<b>Issue #:</b>	N/A
<b>Execution Date:</b>	8.4/2018
<b>Build Version:</b>	Git.
<b>Notes:</b>	Feature not fully developed yet.
<b>Tester Name:</b>	Ahmed Essam

<b>TC_ID</b> ■ 4	<b>Test Case Name</b> ■ Enemies can chase the player
<b>Purpose:</b>	Ensure that the basic functionalities of the enemy is working.
<b>Prerequisite:</b>	- The player started a game level. - The player stands in front of the enemy.
<b>Priority:</b>	<b>High Priority</b>
<b>Steps:</b>	1 – Start the game 2 – Choose a level. 3 – Approach an enemy. 4 – Make sure that there is a line that connects the player with the enemy and that isn't obstructed by a wall.
<b>Expected Results:</b>	The enemy shall start chasing the player.
<b>Actual Results:</b>	The enemy did start chasing the player.
<b>State:</b>	Pass.
<b>Issue #:</b>	N/A
<b>Execution Date:</b>	8.4/2018
<b>Build Version:</b>	Git.
<b>Notes:</b>	Enemy movement is clumsy.
<b>Tester Name:</b>	Ahmed Essam

<b>TC_ID</b> ■ 5	<b>Test Case Name</b> ■ Enemies should patrol their surroundings if not engaged
<b>Purpose:</b>	Ensure that the basic functionalities of the enemy is working.
<b>Prerequisite:</b>	The player started a game level, and the level has enemies in it.
<b>Priority:</b>	<b>Medium Priority</b>
<b>Steps:</b>	1 – Start the game

Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

	2 – Choose a level that has enemies in it. 3 – Observe enemies' movements.
<b>Expected Results:</b>	Enemies should patrol their surroundings.
<b>Actual Results:</b>	Enemies did patrol their surroundings.
<b>State:</b>	Pass.
<b>Issue #:</b>	N/A
<b>Execution Date:</b>	8.4/2018
<b>Build Version:</b>	Git.
<b>Notes:</b>	When enemies start chasing the player, they never let him go and return to their paths.
<b>Tester Name:</b>	Ahmed Essam

## 2.3 Non-functional Testing

### 2.3.1 Performance Testing

We tested the application on the following Android devices:

- Samsung Galaxy Ace 7,                      - Samsung Galaxy Note 3,
- Samsung Galaxy Note 2,                      - Samsung S6.

The RAM usage was in the range of 50 Mbs to 60 Mbs, and the game ran smoothly on at least 55 fps. It was also tested on the following laptops:

- Lenovo Z51-70,                      - HP Envy 15-k201n.

Memory usage on desktop fluctuated from 60 Mbs of RAM to 140 Mbs of RAM, and it ran smoothly on 60 fps.



Geometry Adventure	CM-identifier: STP-1.0
Software Test Plan	Date: 8/4/2018

### 3. References

<https://reqtest.com/testing-blog/sample-test-plan/>

<https://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty>

### 4. Glossary

JUnit: a unit testing framework for the Java programming language.

Git: a version control system for tracking changes in computer files and coordinating work on those files among multiple people.

ECS: Entity-component-system architectural pattern.

fps: Frames per second

RAM: Random Access Memory

Mbs: Megabytes