NIIT Unity Developer Test

## Unity version

* 2019.3

## Expected time to complete

* 2 days

## Items to return

* A zipped-up Unity project folder with your changes
* All documentation containing answers, notes and comments made during the test.

# Introduction

Welcome to the NIIT Unity Developer test.

## Background

Our internally developed comedy platformer "Ninj-haha" really has the client impressed but the developer who created it left the company last week before the project was complete. There are various things wrong with it and a new feature that still needs adding. Can you get the game back on track?

When submitting your project, please include any documentation containing answers and notes that you made during the test. It is much easier to understand what someone was thinking, and why they have implemented something in a certain way if a few notes are included.

A local Git repository is included in the project directory. Please keep this updated as you see fit.

We hope you have fun!

## Control Scheme

WASD – Moves the Ninja.

## Current Functionality Overview

The Ninja can move around the environment.

The Ninja can pick up stars from the floor to increase the score, pick up health packs to increase his health and can hit spears in the ground to decrease his health.

When moving close to some intractable objects (e.g. campfire), the objects respond to the presence of the Ninja.

# **1. Analysis**

As the Ninja moves around the environment, we wanted him to trigger interactions with certain objects. To facilitate this, a simple interaction system was created.

1. Using the campfire as an example, please give a brief overview of how this system works? **(3 Marks)**

There is Sphere Collider Around the Campfire, so whenever the player character triggers the collider, the fire particle system is enabled.

1. The developer decided to use a well-known design pattern to help implement the interaction system.
   1. Can you identify and describe the design pattern used? **(2 Marks)**

A singleton class is used in this situation.

* 1. Why do you think this design pattern was chosen? **(2 Mark)**

Singletons come into effect only when they are required, in this case the fire effect is required whenever the player is near it, in this way processing power and memory is saved.

# **2. Compile Error**

We tried to run the game earlier today but there was a compile error. Please fix this error so that our QA department can continue to test the game. **(1 Mark)**

The x,y,z positions of an object can be changed directly, instead create a new Vector3 with the new position and assign it to the objects position.

# **3. Navigation Issues**

During some testing, the QA department is reporting that there are multiple issues with the current navigation system. As a result, the following bug reports have been filed. Can you investigate and fix them please?

|  |
| --- |
| Bug Report (2 Marks) |
| Description |
| The Ninja cannot walk or run across the first path in the centre of the map. |
| Expected Results |
| The whole environment should be navigable. |
| Actual Results |
| The central dirt path cannot be navigated. |
| Reproduction Steps |
| 1. Use the WASD controls to navigate around the environment. 2. Try to walk across the dirt path in the centre of the map. |
| Accompanying Files |
| [BugReports\CantRunOnPath.mp4](file:///D:\Projects\GameJam\DeveloperTest\CodeTestDoc\BugReports\Cantrunonpath.mp4) |

Fixed the Issue by making the centre path “walkable” and pond area “not walkable” and baked the navigation mesh again.

|  |
| --- |
| Bug Report (2 Marks) |
| Description |
| Obstacles in the environment do not block the player from walking. |
| Expected Results |
| The Ninja should not be able to walk through the pond, trees, walls etc. |
| Actual Results |
| The Ninja character can walk straight through objects that should block his way. |
| Reproduction Steps |
| 1. Use the WASD controls to navigate around the environment. 2. Walk towards an object that should block navigation (i.e. the pond, wall, tree). 3. Walk straight through the object. |
| Accompanying Files |
| [BugReports\CanRunThroughWalls.mp4](file:///D:\Projects\GameJam\DeveloperTest\CodeTestDoc\BugReports\CanRunThroughWalls.mp4) |

I have added Sphere colliders to Rocks, Capsule Colliders to Trees and Box Colliders to Castel walls, so as to prevent the player from passing through them.

# **4. Health Bar and Character Controller**

The designers asked for a “Hard Mode” to be added where the Ninja starts with half his maximum health. This mode can be toggled using a Boolean value in the game controller. Testers are reporting that this feature is not working and that the Ninjas health is always full when they start playing.

Can you explain why this is happening and fix it? **(3 Marks)**

The Condition where the game is in “Hard Mode” or not was checked in Awake() method, if the condition is checked in Start() Method it resolved the issue.

# **5. Collectable System**

The collectable system is used for star pickups and scoring as well as player hazards and health pickups.

We received the following bug reports for our QA team. Can you investigate and fix these issues?

|  |
| --- |
| Bug Report (2 Marks) |
| Description |
| Each star should only be worth 1 point but every time a star is picked up, 2 is added to the score. |
| Expected Results |
| Only 1 point should be added to the score everything a star is picked up. |
| Actual Results |
| 2 points are being added to the score every time a star is picked up. |
| Reproduction Steps |
| 1. Use the WASD controls to navigate around the environment. 2. Run over to a star. 3. Collect the star by colliding with it. |
| Accompanying Files |
| [BugReports/DoubleScoreStars.mp4](file:///D:\Projects\GameJam\DeveloperTest\CodeTestDoc\BugReports\Doublescorecoins.mp4) |

|  |
| --- |
| Bug Report (3 Marks) |
| Description |
| Stars can be picked up multiple times while they are animating away. |
| Expected Results |
| Only 1 point is added for each star. |
| Actual Results |
| Multiple points can be added for each star. |
| Reproduction Steps |
| 1. Use the WASD controls to navigate around the environment. 2. Run over to a star and collect it. 3. While the star is animating away, move out of the area and back in to score additional point. |
| Accompanying Files |
| [BugReports\DoubleCollectStar.mp4](file:///D:\Projects\GameJam\DeveloperTest\CodeTestDoc\BugReports\Doublecollectcoin.mp4) |

Here the double count is because even after collecting the Collectable items the collider still remains active, hence if we disable the collider after triggering them, the issue will be resolved.

# **6. Refactor**

It has been identified that there is a circular dependency between the “Collectable” classes and the “CollectableController” class.

1. What is a circular dependency and why is this a bad thing? **(3 Marks)**

Consider 2 classes ‘A’ and ‘B’, if A is dependent on B and also B is dependent on A, then there exist a circular dependency. It’s not recommended because it can lead to infinite loop/Deadlock.

1. Update the collectable system to use the observer pattern to remove the circular dependencies. **(5 Marks)**

Here in the new Method a Pickups scripts is attached to the player, whenever the player triggers the pickups, the suitable action is taken for the pickup.

1. Can you explain the following?
   1. Why is this observer pattern solution better than the original? **(1 Mark)**

This new Method removes the circular Dependency and removes the risk for Deadlock.

* 1. Why is the observer pattern a suitable solution to this problem? **(1 Mark)**

This new Method for collectables is much less complex Since there is only one Script Controlling the Picks, which is attached to the Player.

# **7. Inventory System**

The game is playing a lot better now and the QA department is reporting fewer bugs. Now is a good time to add an inventory system to keep track of the items the player has collected.

1. The section of the Game Design Document below describes how the inventory system should work; can you implement a working version the fits these specifications? **(12 Marks)**

Bonus marks for using test driven development and including a test suite in your answer **(2 Marks)**

|  |
| --- |
| Inventory System |
| Overview  The player runs around the environment collecting items that then appear in their inventory.  Requirements   * Needs to keep track of all items collected by the player. * Needs to provide a visual representation of the inventory at any given time.   Features   * Items can be:   + Picked up and added to the inventory.   + Equipped from the inventory into the player’s hand.   + Dropped back onto the environment. * Items of the same type should only occupy a single slot in the inventory. * The total number of items in the inventory should not exceed 100.   Resources Location   * UI – Assets/InventoryItems/Sprites * Models – Assets/InventoryItems/Fbx |

1. The designers have now completed their final draft of the UI for the new inventory system. Can you follow these designs and implement the inventory system UI into the game please? The UI should remain visually consistent across all screen sizes and aspect ratios so bear this is mind. **(6 Marks)**

The Inventory has been Created According to the Requirements, Also the UI is Visually Consistent across all screen Sizes.



