

A QUICK GUIDE TO ARTIFICIAL INTELLIGENCE WITH UNITY

Get started fast with Artificial Intelligence fast.

Patrick Felicia

A QUICK GUIDE TO ARTIFICIAL INTELLIGENCE WITH UNITY

Copyright © 2017 Patrick Felicia

All rights reserved. No part of this book may be reproduced, stored in retrieval systems, or transmitted in any form or by any means, without the prior written permission of the publisher (Patrick Felicia), except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either expressed or implied. Neither the author and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

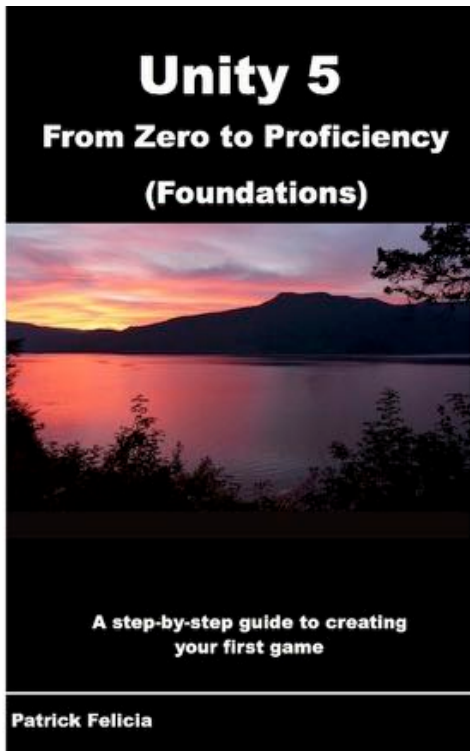
First published: September 2017

Published by Patrick Felicia

CREDITS

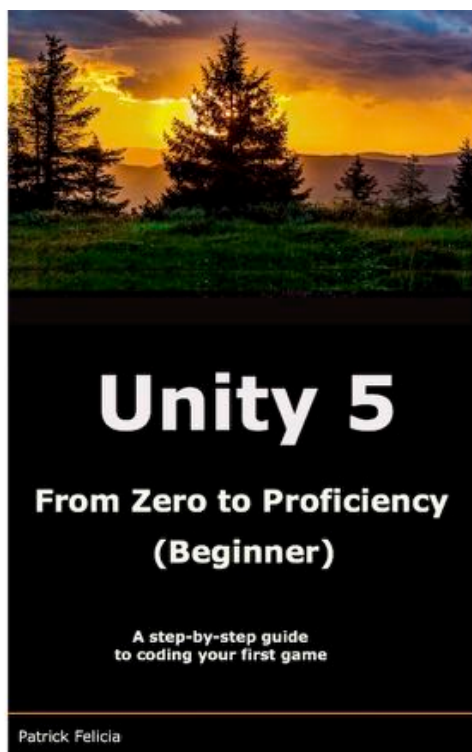
Author: Patrick Felicia

BOOKS FROM THE SAME AUTHOR



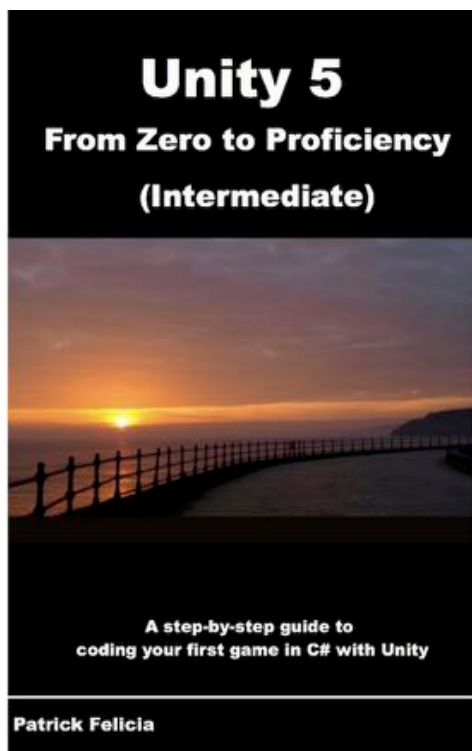
Unity 5 from Zero to Proficiency (Foundations)

In this book, you will become more comfortable with Unity's interface and its core features by creating a project that includes both an indoor and an outdoor environment. This book only covers drag and drop features, so that you are comfortable with Unity's interface before starting to code (in the next book). After completing this book, you will be able to create outdoors environments with terrains and include water, hills, valleys, sky-boxes, use built-in controllers (First- and Third-Person controllers) to walk around the 3D environment and also add and pilot a car and an aircraft.



Unity 5 from Zero to Proficiency (Beginner)

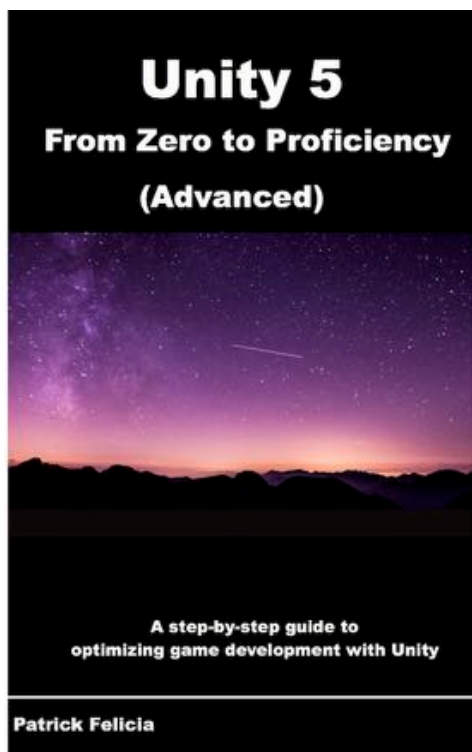
In this book, you will get started with coding using JavaScript. The book provides an introduction to coding for those with no previous programming experience, and it explains how to use JavaScript in order to create an interactive environment. Throughout the book, you will be creating a game, and also implementing the core mechanics through scripting. After completing this book you will be able to write code in JavaScript, understand and apply key programming principles, understand and avoid common coding mistakes, learn and apply best programming practices, and build solid programming skills.



Unity 5 from Zero to Proficiency (Intermediate)

In this book, you improve your coding skills and learn more programming concepts to add more activity to your game while optimizing your code. The book provides an introduction to coding in C#. Throughout the book, you will be creating a game, and also implementing the core mechanics through scripting.

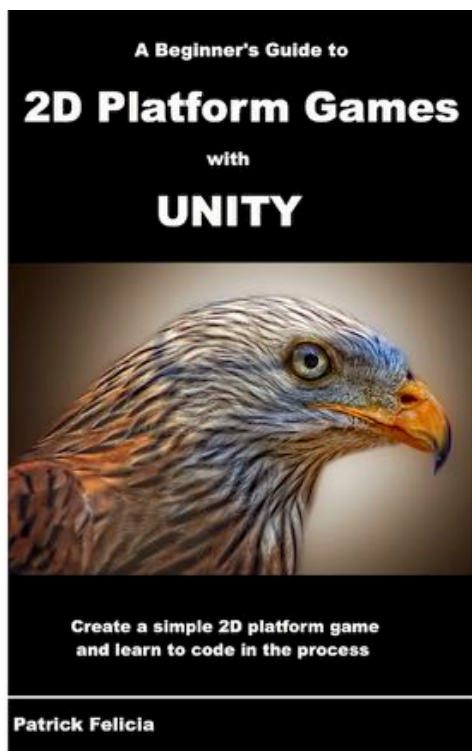
After completing this book you will be able to write code in C#, understand and apply Object-Oriented Programming techniques in C#, create and use your own classes, use Unity's Finite State Machines, and apply intermediate Artificial Intelligence.



Unity 5 from Zero to Proficiency (Advanced)

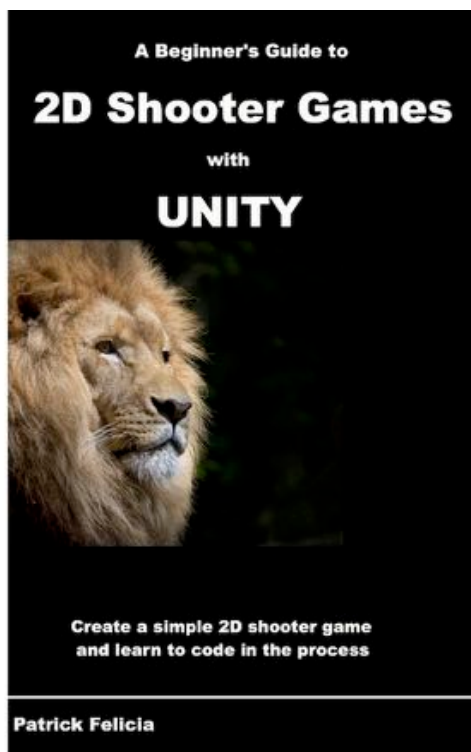
In this book, which is the last in the series, you will go from Intermediate to Advanced and get to work on more specific topics to improve your games and their performances.

After completing this book, you will be able to create a (networked) multi-player game, access Databases from Unity, understand and apply key design, patterns for game development, use your time more efficiently to create games, structure and manage a Unity project efficiently, optimize game performances, optimize the structure of your game, and create levels procedurally.



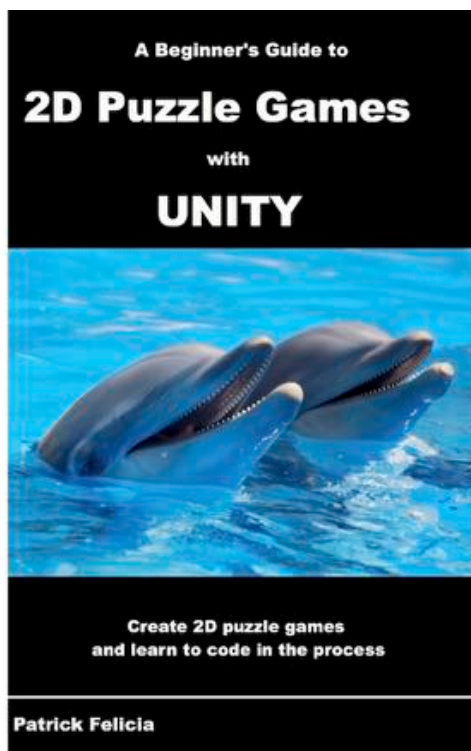
A Beginner's Guide to 2D Platform Games with Unity

In this book, you will get started with creating a simple 2D platform game. The book provides an introduction to platform games , and it explains how to use C# in order to create an interactive environment.



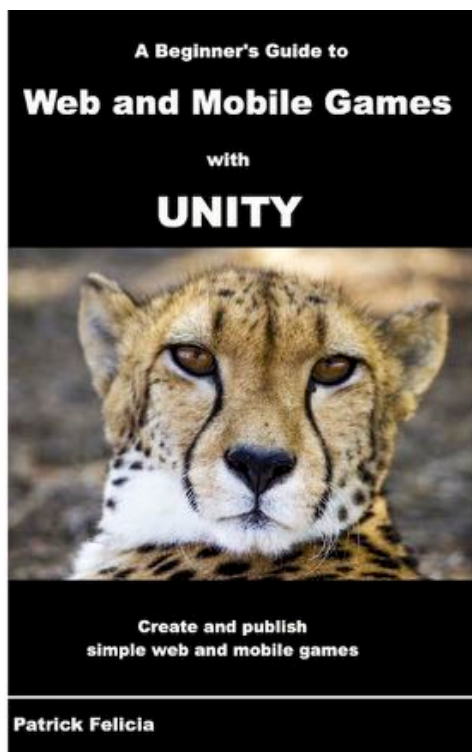
A Beginner's Guide to 2D Shooter Games with Unity

In this book, you will get started with creating a simple 2D shooter game. The book provides an introduction to 2D shooter games, and it explains how to use C# in order to create an interactive environment



A Beginner's Guide to 2D Puzzle Games with Unity

In this book, you will get started with creating four different types of puzzle games. The book provides an introduction to 2D puzzle games , and it explains how to use C# in order to create four additive types of puzzle games including: word games (i.e., hangman), memory game (i.e., simon game), card matching game, and a puzzle.



A Beginner's Guide to Web and Mobile Games with Unity

In this book, you will get started with exporting a simple infinite runner to the web and Android. The book provides an introduction to how to export and share your game with friends on the Web and on Android Play. It provides step-by-step instructions and explains how to easily share a simple game with your friends so that they can play it on your site or an Android device including: processing taps, exporting the game to a web page, debugging your app, signing your app, and much more.

ABOUT THE AUTHOR

Patrick Felicia is a lecturer and researcher at Waterford Institute of Technology, where he teaches and supervises undergraduate and postgraduate students. He obtained his MSc in Multimedia Technology in 2003 and PhD in Computer Science in 2009 from University College Cork, Ireland.

He has published several books and articles on the use of video games for educational purposes, including the Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches (published by IGI), and Digital Games in Schools: a Handbook for Teachers, published by European Schoolnet.

Patrick has published over 10 books on Unity, covering several key skills such as C# and JavaScript in Unity, 3D and 2D game development with Unity, as well as 3D Character Animation.

Patrick is also the Editor-in-chief of the International Journal of Game-Based Learning (IJGBL), and the Conference Director of the [Irish Conference on Game-Based Learning](#), a popular conference on games and learning organized throughout Ireland.

SUPPORT AND RESOURCES FOR THIS BOOK

To complete the activities presented in this book you need to download the startup pack on the companion website; it consists of free resources that you will need to complete your projects, including bonus material that will help you along the way (e.g., cheat sheets, introductory videos, code samples, and much more).

- Please open the following link: <http://learntocreategames.com/books/>
- Select the corresponding book.
- On the new page, click on the link labelled “Book Files”, or scroll down to the bottom of the page.
- In the section called “**Download your Free Resource Pack**”, enter your email address and your first name, and click on the button labeled “**Yes, I want to receive my bonus pack**”.
- After a few seconds, you should receive a link to your free start-up pack.
- When you receive the link, you can download all the resources to your computer.

This book is dedicated to Helena

TABLE OF CONTENTS

1 Creating Simple AI with Built-in Characters	8
Setting up the environment	9
Simple AI (drag and drop) using Unity's built in Ai characters	11
Adding more obstacles	15
Grouping intelligent NPCs	21
Collision detection	22
Off-mesh links	26
Using areas and costs.	29
Making it possible for NPCs to jump	34
2 Frequently Asked Questions	36
Navigation	37
3 Thank you	38

PREFACE

To be able to help people like you, I have designed and published more than 8 books on Unity; these books are in-depth and really provide a significant amount of information on a wide range of topics related to Unity, including 2D/3D game development, Artificial Intelligence, Animation, and much more...

This being said, while these books are comprehensive, many readers, like you, may just want to focus on a particular topic and get started fast.

This book is part of a series entitled **A Quick Guide To**, and does just this. In this book series, you have the opportunity to get started on a particular topic in less than 60 minutes, delving right into the information that you really need. Of course, you can, after reading this book, move-on to more comprehensive books; however, I understand that sometimes you may have little time to complete a project and that you need to get comfortable with a topic fast.

In this book entitled “**A Quick Guide to Artificial Intelligence with Unity**” you will discover how to quickly create realistic simple Artificial intelligence for your games with no coding involved, and make it possible for Non-Player Characters to follow the player, avoid obstacles, and only navigate through zones or areas that you have defined.

WHAT YOU NEED TO USE THIS BOOK

To complete the project presented in this book, you only need Unity 5.5 (or a more recent version) and to also ensure that your computer and its operating system comply with Unity's requirements. Unity can be downloaded from the official website (<http://www.unity3d.com/download>), and before downloading it, you can check that your computer is up to scratch on the following page: <http://www.unity3d.com/unity/system-requirements>. At the time of writing this book, the following operating systems are supported by Unity for development: Windows XP (i.e., SP2+, 7 SP1+), Windows 8, and Mac OS X 10.6+. In terms of graphics card, most cards produced after 2004 should be suitable.

In terms of computer skills, all knowledge introduced in this book will assume no prior programming experience from the reader. So for now, you only need to be able to perform common computer tasks, such as downloading items, opening and saving files, be comfortable with dragging and dropping items and typing, and be relatively comfortable with Unity's interface.

Who this Book is for

WHO THIS BOOK IS FOR

If you can answer **yes** to all these questions, then this book is for you:

1. Would you like to learn how to create basic AI (Artificial Intelligence) for your games?
2. Are you already comfortable using Unity's interface?
3. Would you like to discover more features in Unity?
4. Although you may have had some prior exposure to Unity and coding, would you like to discover game AI?

WHO THIS BOOK IS NOT FOR

If you can answer yes to all these questions, then this book is **not** for you:

1. Can you already create basic AI for your games?
2. Are you looking for a reference book on Unity programming?
3. Are you a professional Unity developer?

If you can answer yes to all four questions, you may instead look for the other books in the series on the official website:

<http://www.learntocreategames.com/books>.

IMPROVING THE BOOK

Although great care was taken in checking the content of this book, I am human, and some errors could remain in the book. As a result, it would be great if you could let me know of any issue or error you may have come across in this book, so that it can be solved and the book updated accordingly. To report an error, you can email me (learntocreategames@gmail.com) with the following information:

- Name of the book.
- The page or section where the error was detected.
- Describe the error and also what you think the correction should be.

Once your email is received, the error will be checked, and, in the case of a valid error, it will be corrected and the book page will be updated to reflect the changes accordingly.

SUPPORTING THE AUTHOR

A lot of work has gone into this book and it is the fruit of long hours of preparation, brainstorming, and finally writing. As a result, I would ask that you do not distribute any illegal copies of this book.

This means that if a friend wants a copy of this book, s/he will have to buy it through the official channels (i.e., through Amazon, lulu.com, or the book's official website: <http://www.learntocreategames.com/books>).

If some of your friends are interested in the book, you can refer them to the book's official website (<http://www.learntocreategames.com/books>) where they can either buy the book, enter a monthly draw to be in for a chance of receiving a free copy of the book, or to be notified of future promotional offers.

1

CREATING SIMPLE AI WITH BUILT-IN CHARACTERS

In this section, we will start by creating simple AI using Unity's built-in prefabs, including:

- Simple AI (drag and drop) for Unity's built in AI characters.
- Group AI.
- Collision detection.
- Off-mesh links.
- Costs.

So, after completing this chapter, you will be able to:

- Create an NPC that follows the player.
- Create several NPCs with the same behavior.
- Detect collision between the NPC and the player.
- Use off-mesh links and costs for the NPCs' navigation.

SETTING UP THE ENVIRONMENT

In this section, we will set-up the environment; it will consist of:

- A scene with a ground (i.e., a scaled box).
- A first person-controller for the player.
- Non Player Characters (NPCs).

Let's get started:

- Please open Unity and create a new project.
- Once Unity is open, import the **Character** package by selecting: **Assets | Import Package | Characters** from the top menu.
- Once this is done, the following window will appear.

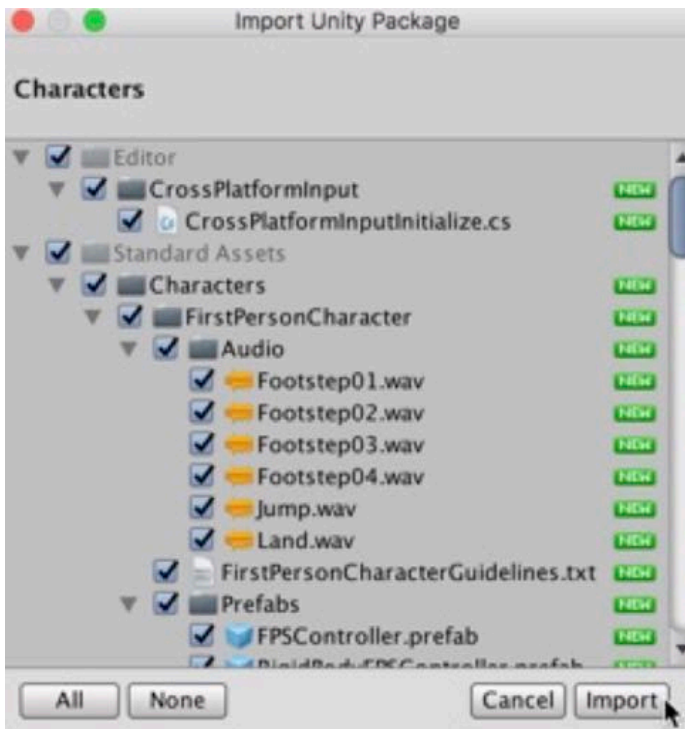


Figure 1: Importing the Character package

Creating Simple AI with Built-in Characters

- You can click on **Import** to import the entire package.
- After a few seconds, this should create a folder called **Characters** located within the folder called **Standard Assets**, as illustrated in the next figure.

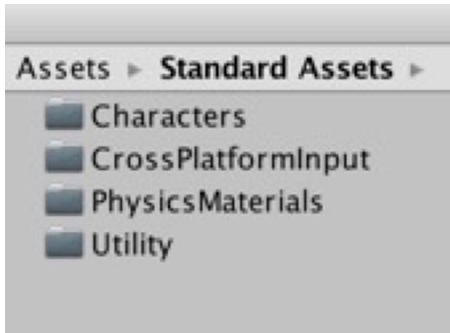


Figure 2: The Characters folder

This package includes the prefabs (or templates) that we will be able to use to implement our first-person character and the NPCs.

Once this is done, we can start to build our environment and add NPCs to it:

- Please create a new cube (**GameObject | 3D Object | Cube**); this will create a new object called **Cube** in the **Hierarchy** window.
- Using the **Hierarchy** window, rename this object **ground**.
- Using the **Inspector**, modify its scale to **(100, 1, 100)**.

SIMPLE AI (DRAG AND DROP) USING UNITY'S BUILT IN AI CHARACTERS

Once the basic set-up is complete, we can now start to add our player and the NPCs.

- Please drag and drop the prefab called **FPSController** from the **Project** folder (within the folder called **Standard Assets | Characters | FirstPersonCharacter | Prefabs**) to the **Scene** view.
- It will create an object called **FPSController**.
- Change the position of this object to **(8, 0, -4)**.
- Please drag and drop the prefab called **AIThirdPersonController** from the **Project** folder (i.e., from within the folder **Standard Assets | Characters | ThirdPersonCharacter | Prefabs**) to the **Scene** view.
- Change its position to **(16, 0, -11)**.

Once you have added these two characters, you can test the scene (i.e., press **CTRL + P**) and walk toward the third person controller; you should see that it is immobile, as per the next figure.



Figure 3: The NPC

What we'd like to do now, is to add the ability for the NPC to follow the player. This will involve setting a target for this NPC.

- Please select the **AIThirdPersonController** object in the **Hierarchy**.

Creating Simple AI with Built-in Characters

- In the **Inspector** window, scroll down to the section (i.e., component) called **AICharacterControl**.

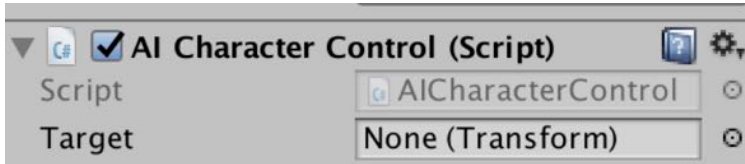


Figure 4: Setting the target (part 1)

- Drag and drop the object **FPSController** to the variable **target**.

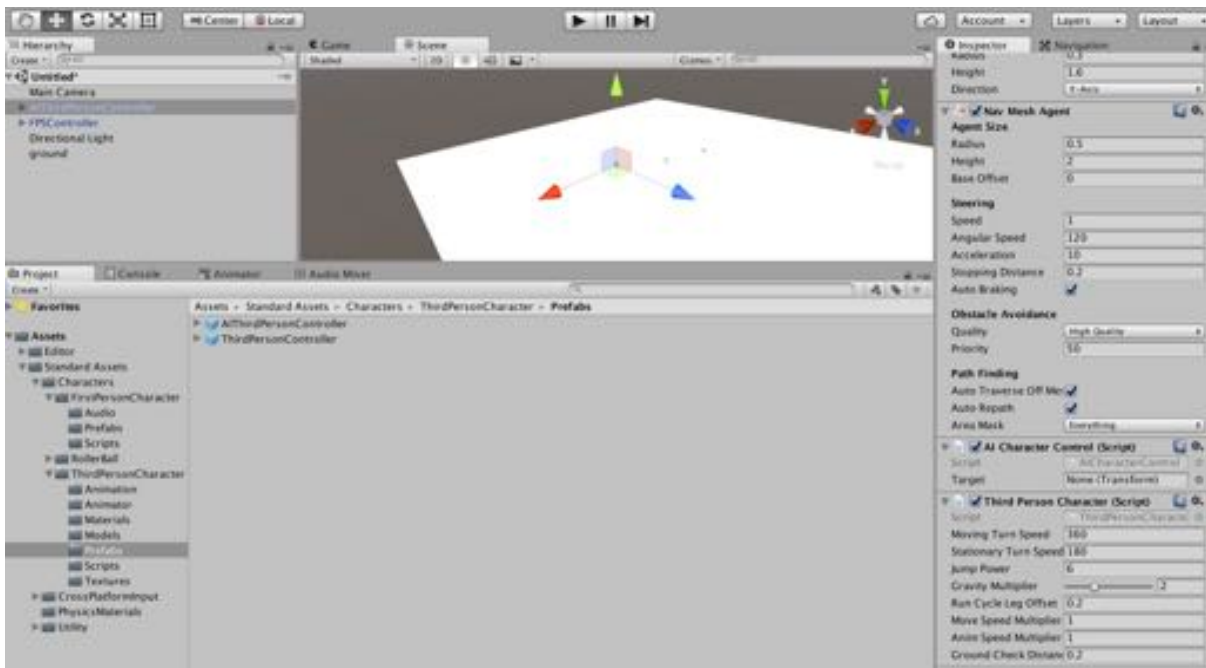


Figure 5: Setting the target (part 2)

- The **AI Character Control** component should then look as in the next figure.



Figure 6: Setting the target (part 3)

- So, by performing this action, we have effectively specified that the target of this AI Agent (i.e., NPC) is the player (i.e., the object **FPSController**).

Next, we need to specify how the agent (i.e., the NPC) can navigate to its target; this will be done using the **Navigation** window.

- Please select the object called **ground**.
- Open the window called **Navigation** (i.e., **Window | Navigation**).
- Click on the tab called **Object**.
- Check the box called **Navigation Static** and leave the **Navigation Area** option to Walkable area.



Figure 7: Specifying the navigation type

- You can then click on the tab called **Bake**, and click on the button labelled **Bake**, within this tab.



Figure 8: Baking the scene

Creating Simple AI with Built-in Characters

By baking the scene, you make it possible for Unity to define and calculate walkable (and static) areas in the scene that the NPC can take to reach its target; without baking the scene, the NPC will not be able to find its way to its target; so this step is very important.

Once the baking process is complete, and if you look at the **Scene** view from above (e.g., along the y-axis), you should see that the ground has turned to blue; indicating that walkable areas have been calculated for this scene.

- You can now play the scene, and you should see that the NPC is now following you.

ADDING MORE OBSTACLES

Now that the NPC can walk towards its target, we will start to add a few obstacles to our scene so that the NPC needs to avoid them to get to the player. These will consist of boxes that we will rescale.

- Please create a new cube (i.e., **GameObject** | **3D Object** | **Cube**) and rename it **wall**.
- Change its scale to **(10,1,10)** and its position to **(0,0,0)**.

Next we will create a new material for this wall so that it can be seen more easily.

- From the **Project** window, select: **Create** | **Material**.
- Rename the new material **red**.
- Select this material (i.e., **red**).
- Using the **Inspector** window, change its colour to red

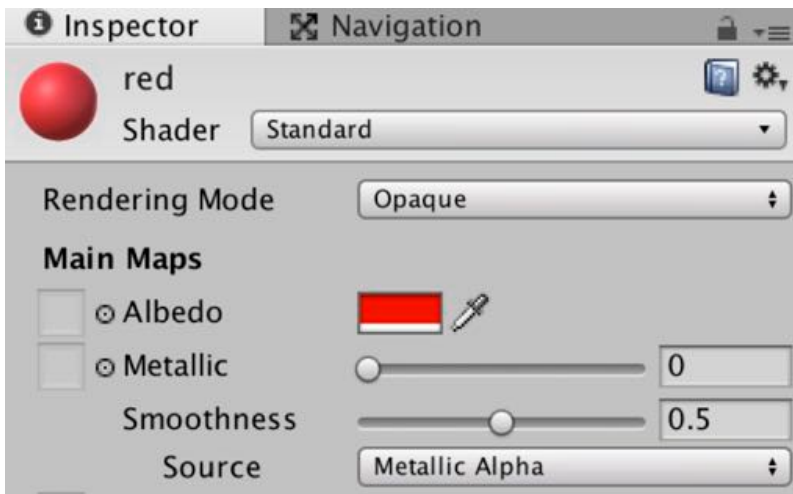


Figure 9: Creating a new material

- Once this is done, you can drag and drop this material (i.e., **red**) on the object called **wall**.
- You can then duplicate this **wall** object four times (i.e., **CTRL + D**) to create a layout similar to the next figure.

Creating Simple AI with Built-in Characters

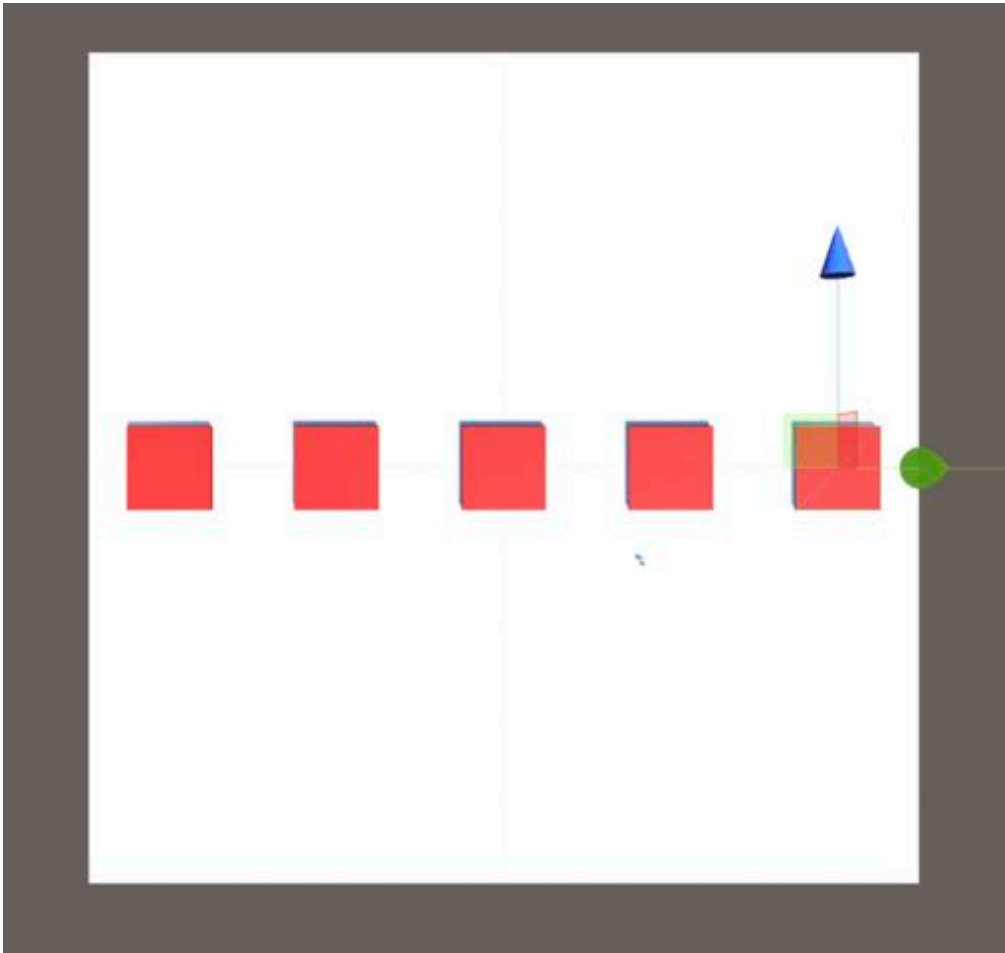


Figure 10: Creating a new layout for the scene

- In the previous example, the walls have the same y and z coordinates, but their x coordinate is respectively **-40**, **-20**, **0**, **20**, and **40**.

You can now duplicate this row 4 times as follows:

- Select the five walls in the **Hierarchy**.
- Duplicate them (**CTRL + D**).
- Using the move tool, move the duplicates so that their z coordinate is **20**.
- Repeat the previous steps three times so as to obtain the following layout (i.e., a grid of 5 by 5 boxes).

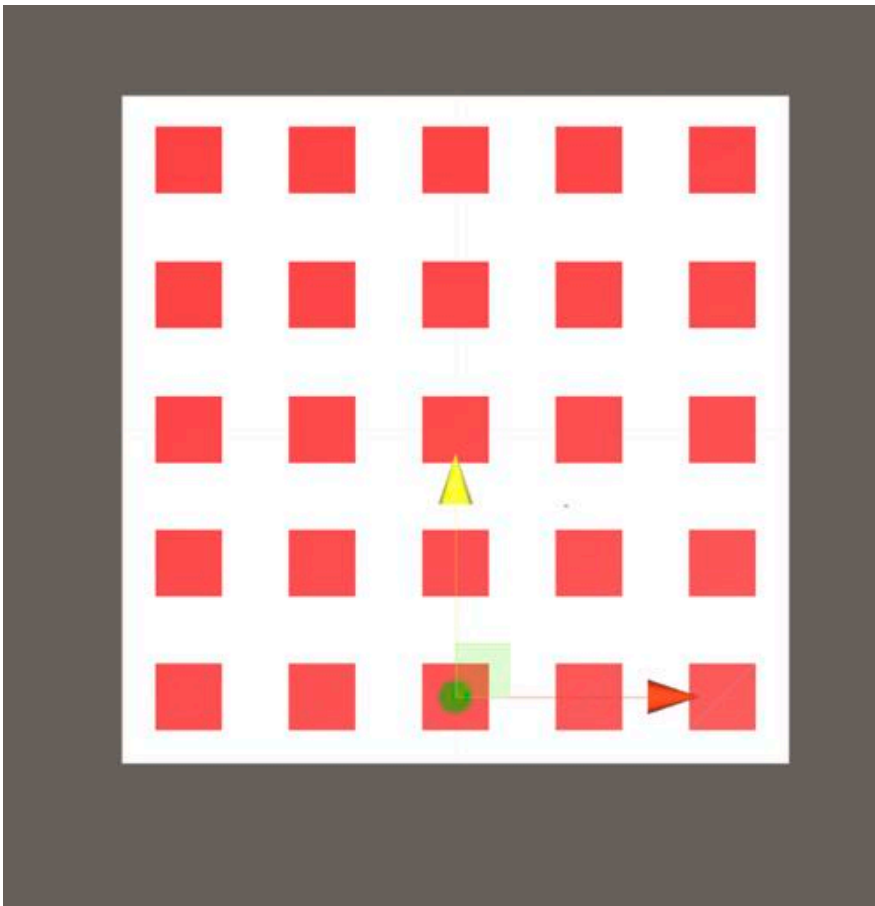


Figure 11: Layout for the scene

Once this is done, we just need to ensure that these walls will be avoided by the NPC when walking towards the player. This will be done by baking the scene again and by including the new walls in the baking process.

- Please select all the walls in the **Hierarchy** window (**Click + Shift key or CTRL + Click**).
- Open the **Navigation** window.
- It should display that you have selected the ground mesh and 25 other objects selected (i.e., the walls).

Creating Simple AI with Built-in Characters

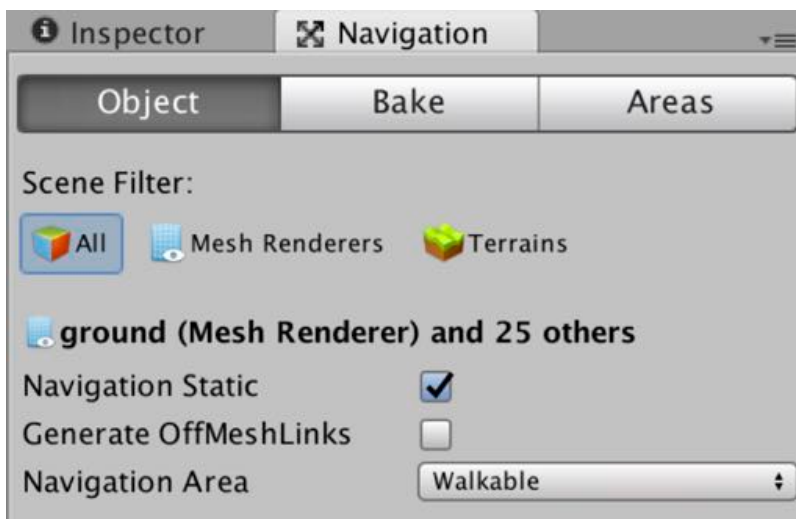


Figure 12: Baking the walls

- As we have done before click on the options **Navigation Static**, and then click on the button labelled **Bake**, that is within the tab called **Bake**.
- After baking the scene, it should look as in the following figure:

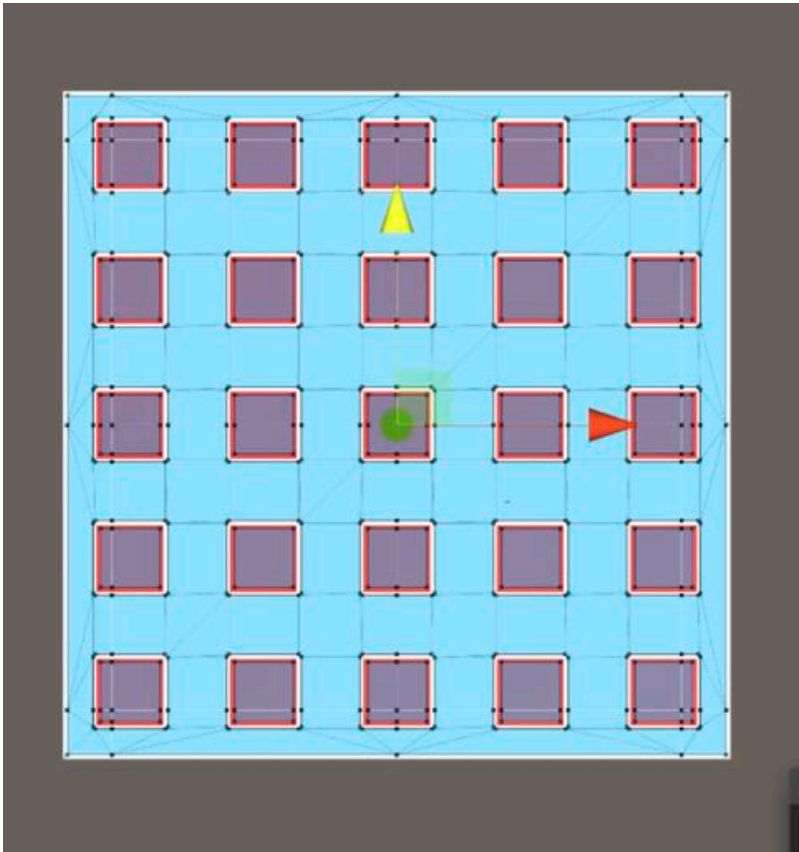


Figure 13: The scene baked

- For a more dramatic effect, and to really see the NPC moving towards the player, you can change the position of the player to **(-40, 0, -10)**.
- As you play the scene, you should see that the NPC follows you by avoiding the walls, as in the next figure.

Creating Simple AI with Built-in Characters

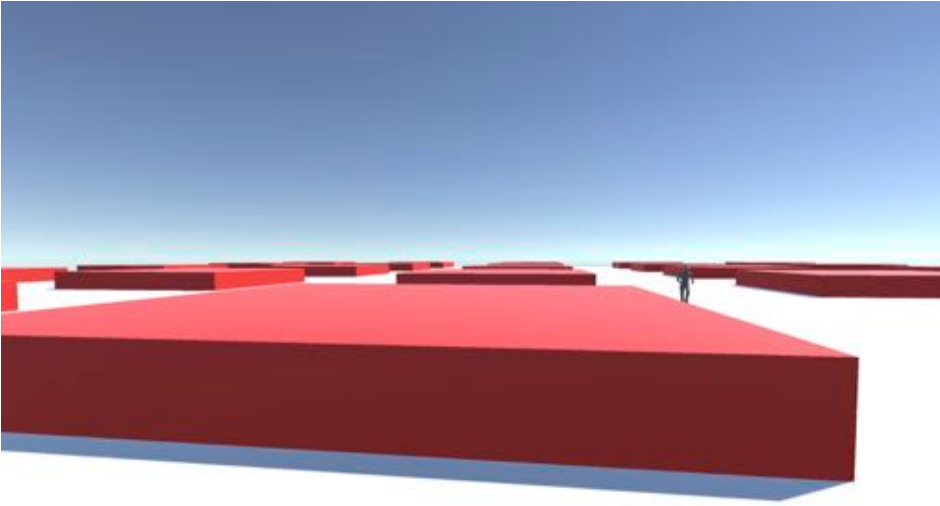


Figure 14: The NPC avoiding the walls to reach the player

GROUPING INTELLIGENT NPCs

Now, to add more challenge to the game, we could add several NPCs in different locations that will all be looking for and following the player. To do so, we just need to duplicate the current NPC and proceed as follows:

- Please change the name of the object **AIThirdPersonController** to **NPC**.
- Drag and drop it (the object called **NPC**) to the **Project** window.
- This will create a prefab called **NPC** (i.e., a reusable template).
- Drag the prefab **NPC** twice to the **Scene** view.
- Change the positions of the duplicates to **(28, 0, 40)** and **(28, 0, -40)**.
- Select all three NPCs, as we will change their attributes simultaneously.
- In the **Inspector** window, scroll down to the component **AICharacterControl**.
- Drag and drop the **FPSController** from the **Scene** view to the variable called **target** for the script **AICharacterControl**.
- As you play the scene, you should see that all three NPCs are now following the player.

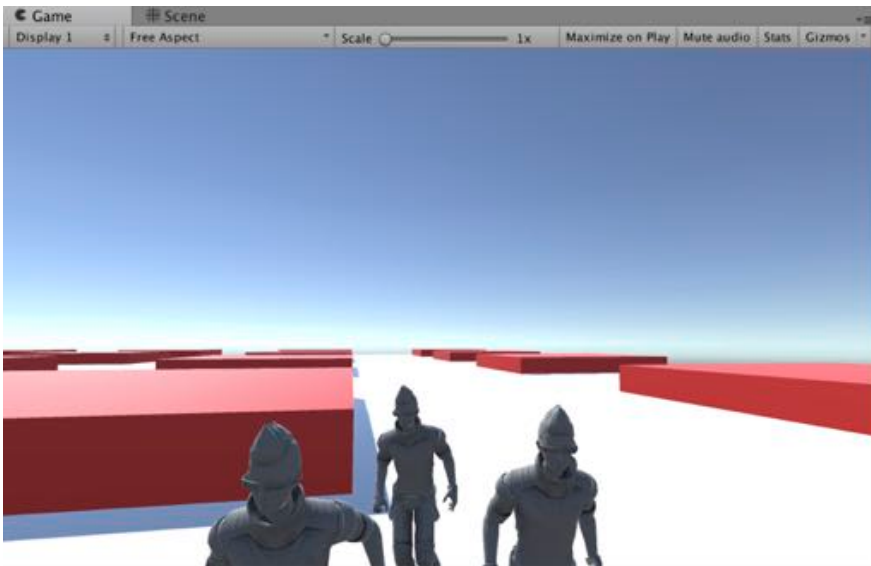


Figure 15: The Three NPCs following the player

Creating Simple AI with Built-in Characters

COLLISION DETECTION

At this stage, we could create a simple rule, whereby the level restarts once one of these NPCs has managed to catch-up with the player; for this we need to detect collision between the player and the NPCs and to reload the current scene in this case. We may also need to decrease the speed of the NPCs, to make the game slightly easier to play.

So let's proceed:

- Please create a new script called **ManageNPC** (from the **Project** window, select **Create | C# Script**).
- Add the following code to it (new code in bold).

```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;
public class ManageNPC : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        GameObject player = GameObject.FindWithTag ("Player");
        float distance = Vector3.Distance (transform.position,
player.transform.position);
        if (distance < 1.5)
            SceneManager.LoadScene
(SceneManager.GetActiveScene ().name);

    }
}
```

In the previous script we check whether the NPC is very close to the player (i.e., the object with the tag called **Player**); if this is the case, we reload the active scene.

- Please save your script and drag and drop it to the **NPC** prefab.

- Add the tag “**Player**” to the **player**: select the **FPSController** object, and using the Inspector window, select “Player” as a tag for this object.

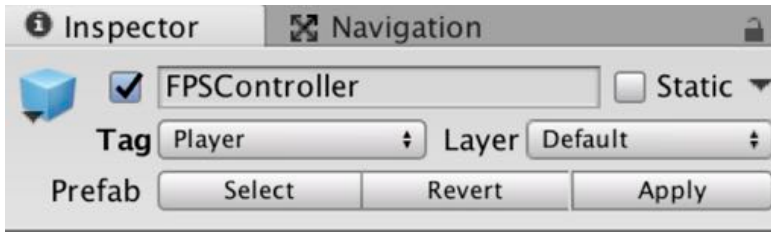


Figure 16: Applying a tag to the player

Now that we have configured the NPC and the player, we just need to ensure that it is possible to reload the current scene, if need be.

- Please save your scene as **chapter1** (i.e., **File | Save Scene As...**).
- Open the **Build Settings** (i.e., **File | Build Settings**).
- Click on the button labelled **Add Open Scene**, as illustrated in the next figure. This will ensure that the current scene is part of the build, and that, therefore, it can be (re)loaded.

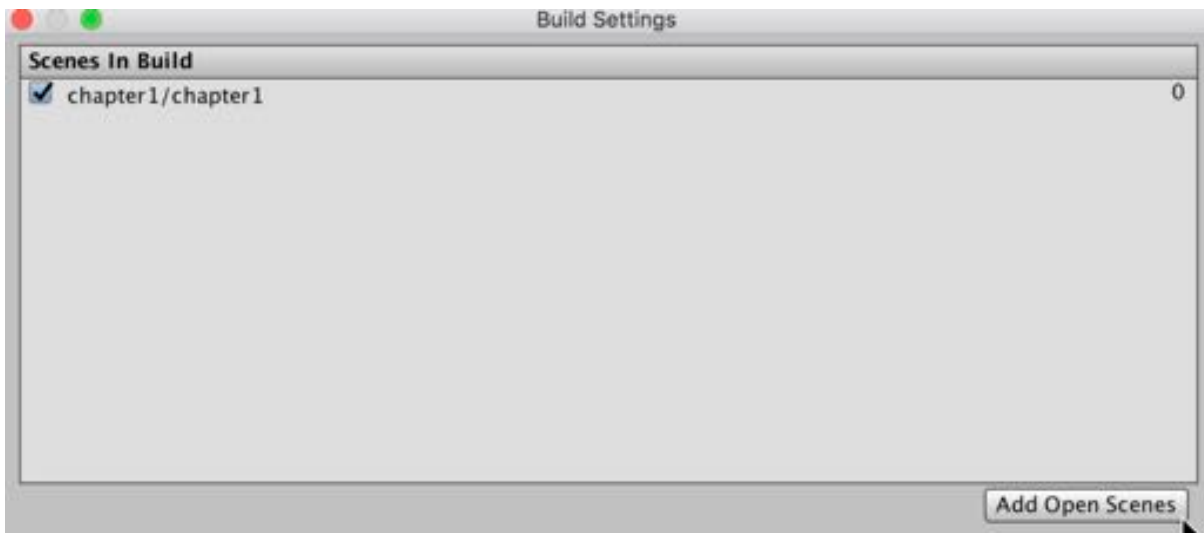


Figure 17: Adding the scene to the Build Settings

Once this is done, you can play the scene. You will notice that, if one of the NPCs is very close, the scene will automatically restart.

Creating Simple AI with Built-in Characters

Note that if the lighting of the game after reloading the scene seems different you can do the following:

- Add a directional light to the scene (**GameObject | Light | Directional Light**).
- Change its rotation to **(90, 0, 0)**.
- Open the Lighting settings (**Window | Lighting**).
- Open the tab called **Scene**.
- Uncheck the box called **Auto**.

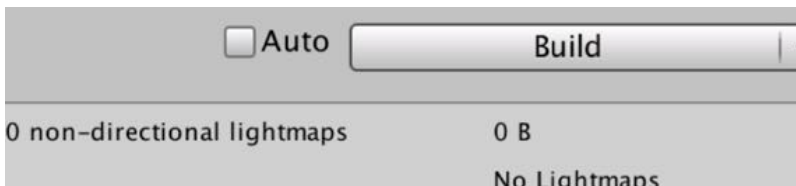


Figure 18: Modifying the light settings

Last but not least, we will decrease the speed of the NPCs:

- Please select the prefab called **NPC**.
- Open the **Inspector** window.
- Scroll down to the component called **ThirdPersonCharacter**.
- Change the attribute **Move Speed Multiplier** to **.7**.

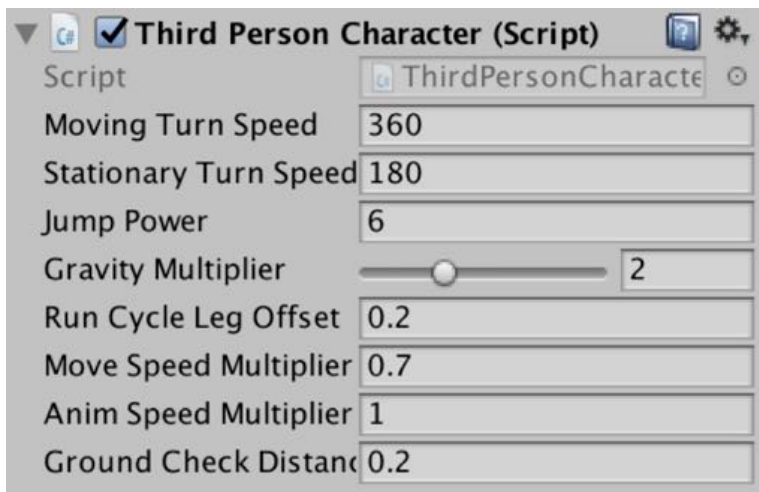


Figure 19: Modifying the speed of the NPCs

You can now test the scene and check that the NPCs are less fast.

OFF-MESH LINKS

So far we have managed to create (**Navmesh**) navigation for a continuous flat surface; however, it would be interesting to introduce a few gaps, and force the NPCs to jump, if need be, to be able to follow the player; for this purpose, we will be using off-mesh links; the idea behind off-mesh links is to provide the NPCs with a way to navigate between two walkable surfaces that may not be connected (e.g., two platforms), by specifying the maximum distance that they can jump to go from the first one to the second one.

So let's proceed:

- Please duplicate the **ground** object.
- Rename the duplicate **ground2**.
- Change its position to **(-75, -3, 0)** and its scale to **(50, 1, 100)**.

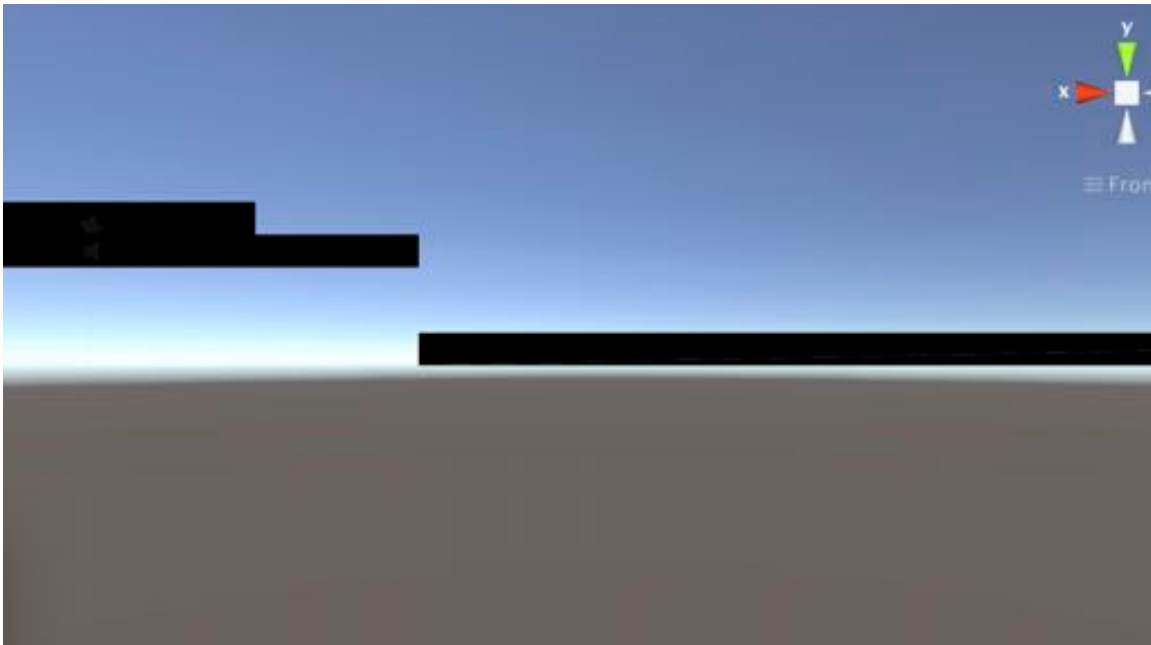


Figure 20: Creating the lower level

Once this is done, we will make sure that we allow the NPC to jump from the upper platform to the lower platform if need be:

- Select both the objects **ground** and **ground2** in the **Scene** view.
- In the **Navigation** window, select the option to **Generate OffMesh Links**.



Figure 21: Setting-up navigation for both levels

- Then click on the **Bake** tab.
- Specify a **Drop Height** of 4.



Figure 22: Modifying the jumping height

- You can then click on the **Bake** button.
- If you open the **Navigation** window, and then look closely at the **Scene** view, you should now see that Unity has generated links between the two levels, as illustrated in the next figure.

Creating Simple AI with Built-in Characters

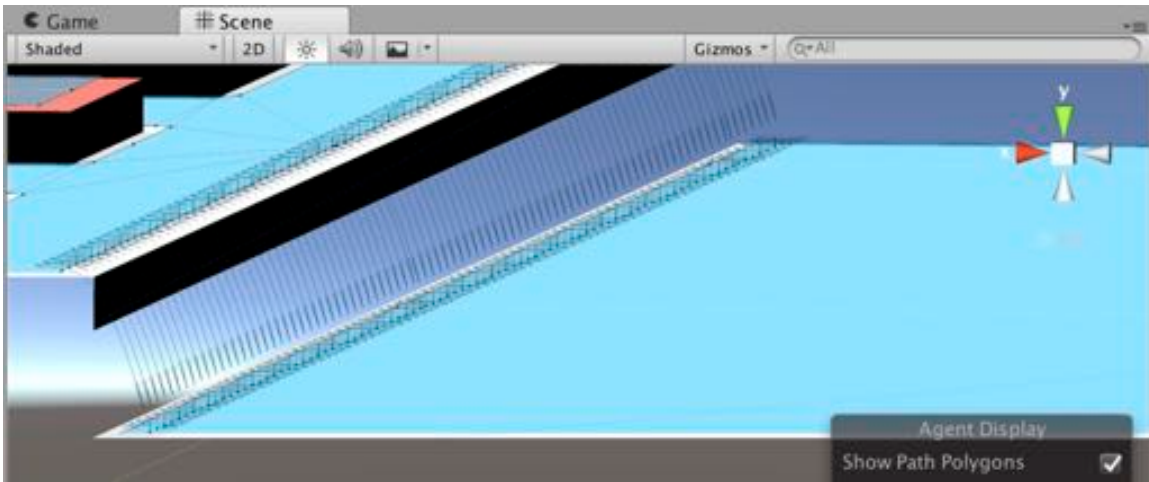


Figure 23: Off-mesh links

In order to test these changes (i.e., ability to jump from platforms for the NPCs), we will modify the color of the bottom level to blue:

- Please create a new **blue** material as we have done previously.
- Apply it to the object called **ground2**.

If you play the scene, jump on the blue area and look back, you should see that the **NPCs** will also jump on your current platform to navigate towards you.

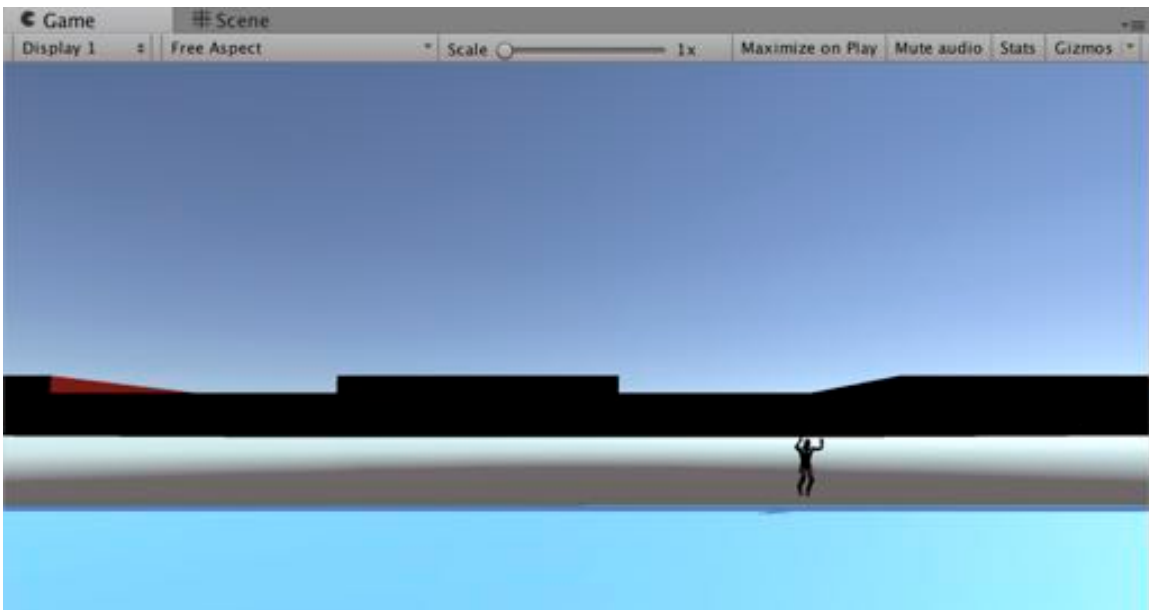


Figure 24: NPCs following Navmesh links

USING AREAS AND COSTS.

In this section, we simulate the presence of water. Although our NPCs may be able to walk through water, this environment may slow-down their progression; so the idea here is to assign costs to different environments so that the NPCs can plan their navigation accordingly and avoid areas with high costs (i.e., that might slow them down too much) in order to reach their destination faster.

First let's add a few more elements to our environment:

- Please create a new cube and rename it **bridge**.
- Change its position to **(-65, 0, 12)** and its scale to **(30, 1, 3)**.
- Create another cube and rename it **platform**.
- Change its position to **(-95, 0, 12)** and its scale to **(30, 1, 30)**.

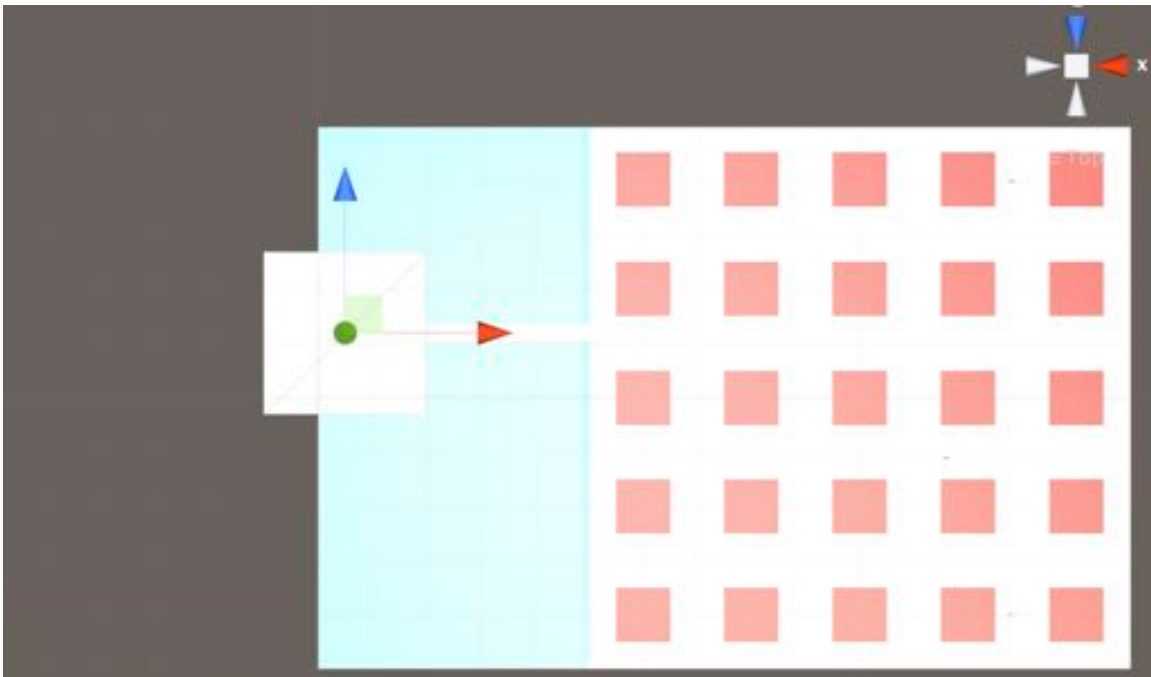


Figure 25: The scene layout

We can then define an area and its associated (navigation) cost:

- Please select the **Navigation** window.

Creating Simple AI with Built-in Characters

- Create a new area called **Water** with a cost of 10.





	Name	Cost
 Built-in 0	Walkable	1
 Built-in 1	Not Walkable	1
 Built-in 2	Jump	2
 User 3	Water	10

Figure 26: Creating a new area

As you can see, this type of area will have a higher cost than the walkable areas that were created by default; so if the NPC has to choose between these two, it should opt for the walkable area, as its cost is lower, and the NPC would therefore arrive to its destination faster.

- Please select the object called **ground2**.
- Using the **Navigation** window set its **Navigation Area** attribute to **Water**.

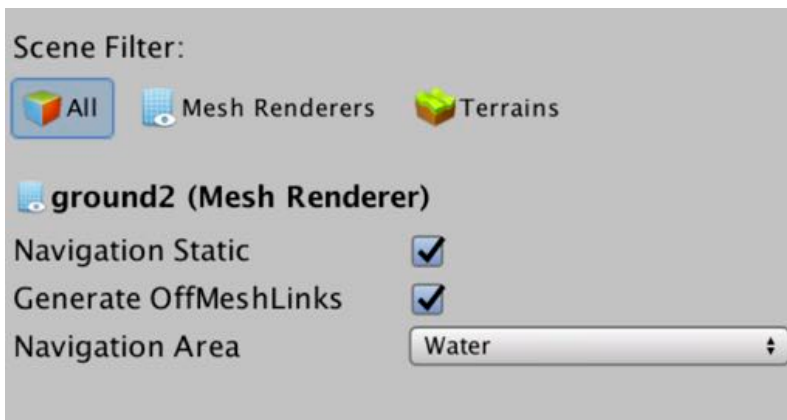


Figure 27: Setting Navigation settings

- Select the objects **bridge**, **platform**, and **ground**.
- Using the **Navigation** window set their attributes as in the next figure:



Figure 28: Setting the navigation (part 2)

- You can now bake your scene.

As you play the scene, you should see that, if you walk on the bridge and look back, the NPCs will also follow the bridge and not jump into the water (as they would have done before).

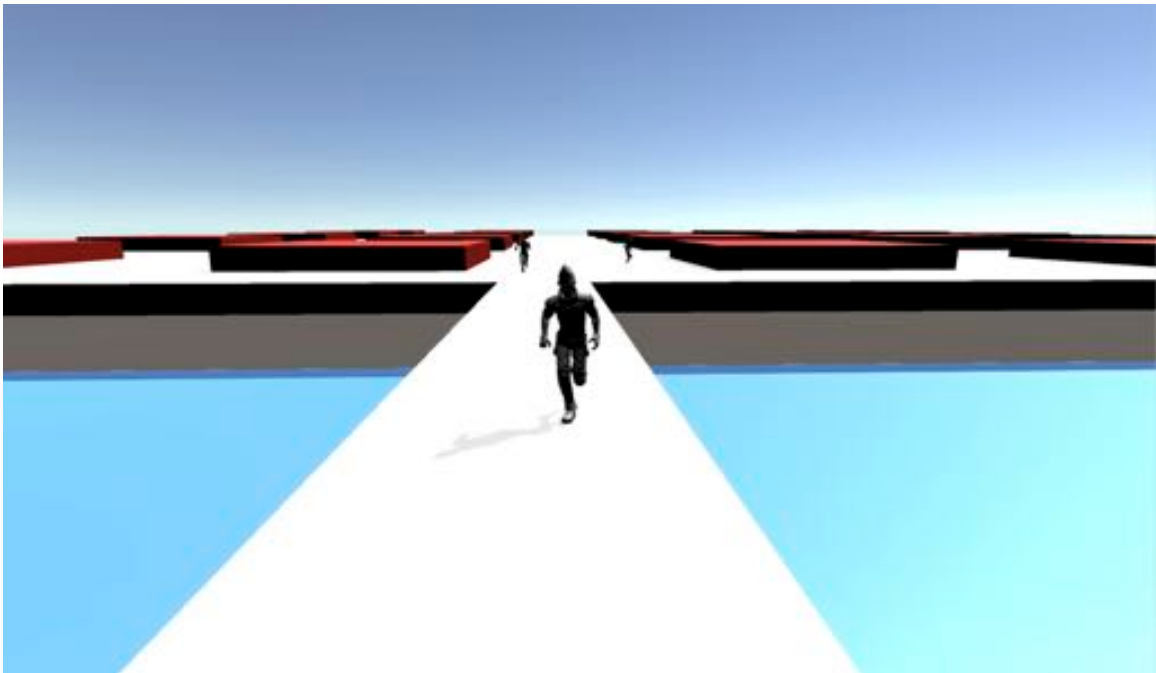


Figure 29: NPC navigation based on costs

Creating Simple AI with Built-in Characters

Note that we could have achieved a similar effect by allowing the NPCs not to walk on the water area; this can be achieved as follows:

Select the NPCs:

- Using the **Inspector** window, scroll down to the section called **NavMeshAgent**.
- Modify the area mask so that **Water** is not ticked in the list, as per the next figure.



Figure 30: Setting-up the area mask

- For testing purposes, you can set the cost for the water to **1**, and play the scene, so that the changes in the NPC navigation are only linked to the area mask and not the cost.

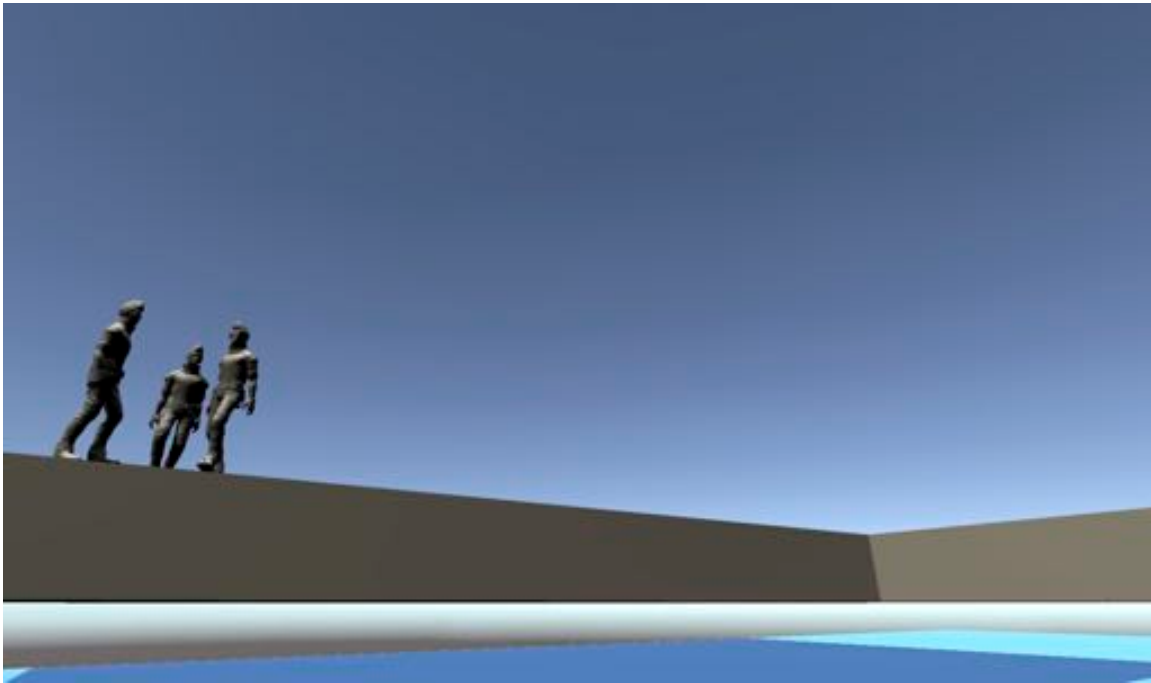


Figure 31: NPCs using area masks

- If you walk in the water, you will see that the NPC will not follow you. This is because the area called water (whether the play is currently) is not part of the area mask for the NPCs.

MAKING IT POSSIBLE FOR NPCs TO JUMP

In this last section, we will add a feature whereby the NPC are able to jump between two platforms. Whereas the NPCs could previously jump vertically to reach a lower level, we will now make it possible for the NPCs to jump horizontally instead.

- Please duplicate the object called **platform** and rename the duplicate **platform2**.
- Change its x coordinate to **-28**.
- Select the two objects **platform** and **platform2**.
- In the **Inspector** window, using the **Object** tab, change the settings as follows:

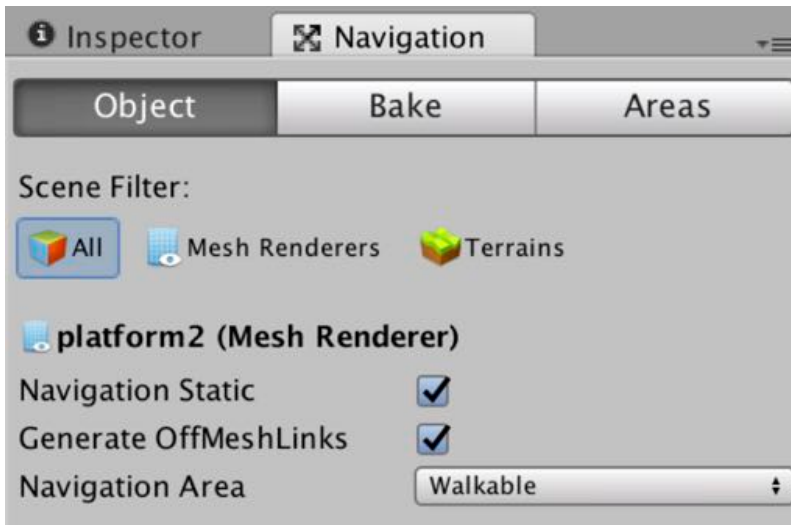


Figure 32: Modifying the Navigation settings

- Click on the **Bake** Tab.
- Change the **Jump Distance** to **4**.



Figure 33: Setting the jump distance

- Bake the scene.

After pressing making sure that the **Navigation** window is active, you should see the following: off-mesh links, symbolized as arrows, have been created between the two platforms.

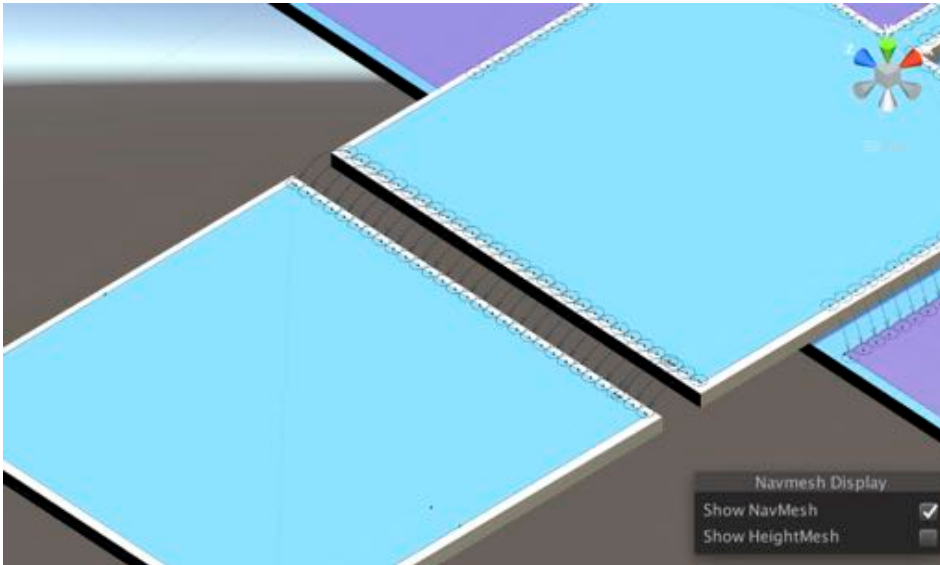


Figure 34: Off-mesh links between the two platforms

If you play the scene, and reach the second platform, you should see that not only do the NPCs use the bridge to reach the player, but they also jump between the two platforms, as per the next figure.

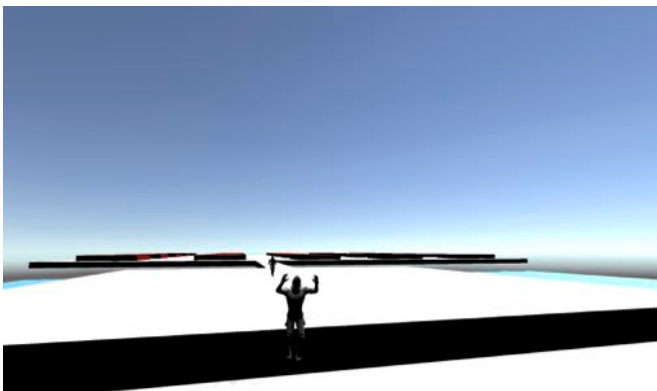


Figure 35: NPC jumping between platforms

2

FREQUENTLY ASKED QUESTIONS

This chapter provides answers to the most frequently asked questions about the features that we have covered in this book. Please also note that some videos are also available on the companion site to help you with some of the concepts covered in this book.

NAVIGATION

Can I create AI in Unity without coding?

Yes, you can, also this may be limited in terms of the features that you can create; this being said, Unity includes built-in packages that you can use to create NPCs that follow the player (or another target) while avoiding obstacles individually or as a group.

Can I make sure that my NPCs avoid some areas?

Yes, you can achieve this using by defining area and associated costs, and by selecting a navigation layer for your NPC. This will ensure that the NPC avoids (or prioritizes) some areas while navigating to its target.

What are waypoints?

Waypoints are temporary targets that are used to provide a goal to your NPC; by adding and combining waypoints, you can define a path (made of successive objectives) that the NPC can follow. These waypoints are then set as successive navigation targets. Note that the waypoints can also be followed in a random sequence, to make the NPC less predictable.

What do I need so that a 3D character starts to navigate to a destination?

To add navigation capability to a 3D character, it will need to include a Navmesh Agent component, and an Animator Controller; you will also need to create a script that will, based on the active state, set a destination for the NPC. Finally, you will need to bake the scene using the **Navigation** window, after selecting objects that the NPC can either avoid or walk on.

Thank you

3

THANK YOU



I would like to thank you for completing this book; I trust that you are now comfortable with the creation of infinite runners. This book is of course only a quick guide to get started with Unity; if you'd like to know more about Unity and 2D games, 3d game or Artificial Intelligence, you may try some of my other books available from the official page: <http://www.learntocreategames.com/books>, especially the book called “[Unity from Proficiency to Mastery \(Artificial Intelligence\)](#)”, as it really provide more in-depth information about AI if you would like to know more about this topic.

So that the book can be constantly improved, I would really appreciate your feedback and hear what you have to say. So, please leave me a helpful review on Amazon letting me know what you thought of the book and also send me an email (learntocreategames@gmail.com) with any suggestion you may have. I read and reply to every email.