Tutorial for 2D Player Movement

1. **Create a new scene.**

Firstly, create a 2D object. Any sprite will do.

Name this 2D object, ‘Player.’

Tag the 2D object as ‘Player in the inspector tab, this will be useful when referring to this object in scripts.

Create another 2D object, this will be named and tagged ‘Floor/Platform’.

1. **Add the Necessary Components**

Add the Rigidbody 2D component to the ‘Player’ object. This means that the ‘Player’ will now be affected by gravity.

Add the Box Collider 2D component to the ‘Player’ and ‘Floor’. This means that the ‘Player’ will now be able to touch/collide with the ‘Floor.’

1. **Create a new script.**

We need to specify what is going to be needed for our player movement. These are going to be our variables.

Our public floats are going to be speed and jump. We will be able to change the values for how fast our ‘Player’ will move and how high our ‘Player’ will jump.

A screenshot of a computer

Description automatically generatedOur one private float is move; this is to determine how our ‘Player’ will move left to right.

We need to access our Rigidbody for our ‘Player’ so a simple line will do to be used as a reference in the script.

1. **Script for the ‘Move’ Variable**

First, assign the ‘move’ variable where we will use the horizontal string name.

A screen shot of a computer code

Description automatically generatedNext, we will assign the velocity of the rigidbody. The velocity for the x axis will be what ever value we put for our ‘speed’ variable multiplied by our ‘move’ variable. We are only concerned about the x-axis because we want our ‘Player’ to only move horizontally not vertically. So, the velocity for the y-axis on our ‘Player’ right now will not change.

1. **Script for the ‘Jump’ Variable**

For our jump variable we want to only focus on our y-axis this time when the space button is pressed, so we only want to add force to our y-axis which will be our ‘jump’ variable. The x-axis will be left unchanged.

A black screen with white text

Description automatically generated

1. **Adding Values**

A screenshot of a game menu

Description automatically generatedNext, we have to add the values for the ‘speed’ and ‘jump’ variable in our inspector tab as well as assigning the rigidbody for our ‘Player.’

Now our ‘Player’ can move from left to right and jump.

Tutorial for Grabbing an Object (Coins)

1. **Create a new 2D object.**

Design the sprite any way you want to.

For this tutorial, a simple yellow circle will work. This will be our coin.

Add tag for our ‘coin’ object named Coins.

Create another 2D object which will be our ‘Player’. And make sure our ‘Player’ object is tagged as ‘Player.’

1. **Add Component**

We want to add the component Circle Collider 2D to our ‘coin’. This allows us to accurately edit the collider around our coin as it is a circle sprite. And we want to add a Box Collider 2D to our ‘Player.

A screenshot of a computer

Description automatically generatedWe want to tick the ‘Is Trigger’ box under the Circle Collider 2D inspector tab. This means that our ‘coin’ will not have a physical collision with other objects, but it still can trigger events when another object collides with it.

1. **Script**

When an object, for example our ‘Player’ touches/collides with our ‘coin’ it will be destroyed by our ‘Player’. This creates the illusion of the ‘Player’ picking up the coin.

A screen shot of a computer program

Description automatically generated

Tutorial for Patrolling Enemy/NPC

1. **Create a new.**

First create a new 2D Object, this will be our ‘enemy.’

Create an empty and name it ‘LeftPoint.’

Create another empty and name it ‘RightPoint.’

Place them apart, this will be the where the ‘enemy’ will move back and forth from.

1. **Adding Components**

For the ‘enemy’, we want to add the components Rigidbody 2D and Box Collider 2D. This will allow our ‘enemy’ to physically collide with other objects such as the ‘floor’ and ‘Player.’

1. **Create new script for ‘enemy’.**

We need to specify what variables we will need in order for our ‘enemy’ to move from the ‘LeftPoint’ to the ‘RightPoint’ and vice versa.

**A computer screen shot of white text

Description automatically generated**

1. **Start Section - Assigning variables.**

We now need to assign our variables of our Rigidbody as well as the current point. We will set the current point to the position of our ‘RightPoint’ under the Start of the script.

A computer screen with white and green text

Description automatically generated

1. **Update Section – ‘RightPoint’**

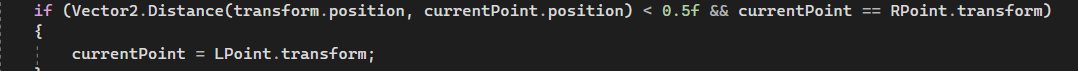
We want our ‘enemy’ to go to a certain direction, we need to make sure that if the current position is our ’RightPoint’ which is true from our Start Section then it will access the velocity of the ‘enemy’ rigidbody’s speed. We only want the ‘enemy’ to move on the x-axis and not the y.

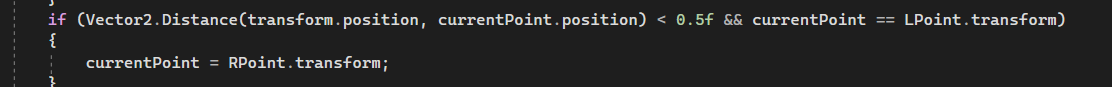
If our current point is anything else such as the ‘LeftPoint’, we want the ‘enemy’ to move in the opposite direction, so we want the speed to be minus what it was going in originally.

A screen shot of a computer program

Description automatically generated

1. **Update Section – ‘LeftPoint’**

Next, we need to be able to set our current point to our ‘LeftPoint’. If the ‘enemy’ has reached the current point (‘RightPoint’) we then want to set the current point to our ‘LeftPoint’.

If we then copy this code and swap our ‘LeftPoint’ and ‘RightPoint’. This will enable our ‘enemy’ to continuously move from our ‘RightPoint’ to our ‘LeftPoint’ and vice versa.

1. **Inspector Tab**

A screenshot of a computer script

Description automatically generatedLastly, we want to assign what is our ‘RightPoint’ and ‘LeftPoint’ in our project. As well as the speed value and the rigidbody of our ‘enemy.’

Tutorial for Player Health and Enemy Damage

1. **Setup**

Create a new 2D Object. Name it ‘Enemy’.

Create another 2D Object and name it ‘Player’. Make sure our ‘Player’ object is tagged as ‘Player.’

1. **Adding Components**

We need to add the component Box Collider 2D to both our ‘Enemy’ and ‘Player’ objects. This will enable both objects to physically touch and trigger the ‘Enemy’ to deal damage to our ‘Player.’

1. **Script for Enemy/NPC Damage**

We need to create a new script for our ‘Enemy.’

Our two variables will be the amount of damage the ‘Enemy’ will do and the player health.

A computer screen shot of a computer code

Description automatically generatedWe want the ‘Enemy’ to only deal damage to our player health when our ‘Player’ and ‘Enemy’ collide with one another.

1. **Script for Player Health**

We need to create a new script for our ‘Player.’

Our two variables for this script will be the ‘Player’s’ maxHealth (which you can set to 10) and health.

Under the start function we want our ‘Player’ health to always be at maxHealth. This means that every time we reload our project, our ‘Player’ will always start at 10 Health.

Any damage done to our ‘Player’ will result in the health value decreasing by the amount of damage that is set. If our ‘Player’ health is equal to or less than 0 this will result in our ‘Player’ being destroyed. In game terms, will result in Player death.

**A computer screen shot of a program code

Description automatically generated**

1. **Assigning in Inspector Tab**

For our Enemy Damage script, we need to specify how much damage the ‘Enemy’ can deal in the inspector tab.

Once we press play in our project, we will see that each time our ‘Player’ and ‘Enemy’ collide the player health will decrease. For example, if we set the damage to 5 and our ‘Player’ health starts at 10. It will take two collisions for our ‘Player’ to be destroyed/killed.

Sources I used:

[Unity 2D Player Move and Jump Tutorial (youtube.com)](https://www.youtube.com/watch?v=6xn0Sokihdc)

[Simple 2D Enemy Patrolling Unity tutorial (youtube.com)](https://www.youtube.com/watch?v=RuvfOl8HhhM&list=LL&index=46)

[Unity 2D Collecting Coins Tutorial (youtube.com)](https://www.youtube.com/watch?v=DZ-3g31jk90&list=LL&index=44)

[Enemies that Deal Damage--Just Add Enemies! (2D Unity Tutorial) (youtube.com)](https://www.youtube.com/watch?v=KF3EVjOhN4c&list=LL&index=14&t=368s)

[Pick Up Items On Collision - Beginner Unity 2D (youtube.com)](https://www.youtube.com/watch?v=2tarnsQGzac&list=LL&index=45)

[How to Make A Simple HEALTH SYSTEM in Unity (youtube.com)](https://www.youtube.com/watch?v=vNL4WYgvwd8&list=LL&index=15)