**Third Person Shooting Player User Guide**

This documentation describes how to use the ShootingPlayer component in your project.

**ShootingPlayer**

Apply this script to a Game object or asset that you want to be your player/want to control. This script is very useful if you want to make either an FPS or Third-Person Shooter games and do not want to use any Unity Assets. This script works better and is already in the Third-Person Shooter mode.

**Behaviours**

* Player Movement
* Player Shooting

**Player Movement**

The character you control does not have the ability to jump, only to move around in the scene. You may apply this script to any Game Object or asset if you want to make a third-person shooter game. The object should be aligned with the z axis pointing forward and the y axis pointing up. It moves the object around using the transform so it should not have a non-kinematic Rididbody attached (a kinematic Rigidbody is fine.)

**Input**

The inputs for this would be the speed, turn smooth time (smooth turning) and setting the controller and Camera. The Speed Input would control how fast the character is able to move, turning smoothly can be adjusted too.

**Player Shooting**

This script works alongside a prefab of the bullet. Each object has a script attached to it: the character game object has two scripts attached – one for its movement (covered above) and the other for its weapon. The Inputs on the weapons script are assigning a fire point (where the bullet comes from) and assigning a bullet prefab (this will shoot projections of that prefab.

The prefab bullet itself has a script attached to it, this makes it travel forward and gives the ability for the bullet to destroy enemy objects. The Inputs that you can change would be its damage and speed. You must assign its Rigidbody to the “rb” Input otherwise it would be useless.

**Sentry Gun**

This documentation describes how to use the Sentry Gun component – this is a form of an object that players can use that will lock on to enemies and start shooting 🡪 it can be destroyed like normal enemies.

Sentry Gun

Apply this script to an object/asset that you wish to be used to destroy enemies. This package will able an object to lock onto enemies and start firing, it can be destroyed and has health like a normal enemy.

Behaviours

* Range
* Lock on
* Shoot
* Be destroyed when health is at 0

**Range** – the range is the area that the sentry gun can see the player in (this is shown through a red sphere in the scene around the object – done by using Gizmos)

**Lock on** – this function targets any object with the tag “Enemy”, to show this visually you must have an empty gameobject at the point you want the turret to rotate (Part to Rotate) as a parent to the shape.

**Shoot** – the sentry gun will have the ability to fire projectiles at enemies (or whatever uses the tag you set) at a fire rate of your choice (in seconds)

**Destroyed** – the enemies the sentry gun shoots will be destroyed along with the bullet that has been shot.

**Inputs**

The components that you may change around in this package:

* **Range** – change the size of the area to make the turret detect enemies further away.
* **Enemy Tag** – change the tag of the object the turret will target.
* **Part to Rotate** – do not change.
* **Turn Speed** – the speed the turret can rotate.
* **Fire Rate** – how many increments (in seconds) the bullets will be shot.
* **Bullet Prefab** – any projectile that you wish the turret to shoot.
* **Fire Point** – like the Part to Rotate, an empty game object that the bullets will spawn at (usually at the tip of the gun barrel)

**Enemy AI**

This documentation describes how to use the Enemy AI component in your project. The AI will actively look for the player when in range and chase/attack the player depending on the range.

This package comes with two scripts you must attach to an empty game object that is a parent of whatever asset/shape you have. The object must have these components (and scripts):

* Nav Mesh Agent
* Capsule Collider – “Is Trigger” off
* Rigidbody

The Scripts:

**Enemy Health** – this script allows the enemy to be killed and destroy its Rigidbody, this only happens when its health reaches 0. You can alter the enemy’s health in the inspector.

**Enemy Controller** – this script controls the enemy’s movements and actions:

* Searching for the player
* Chasing the player when in range
* Attacking the player when in range
* Facing the player

**Player Manager** – put this script on an empty game object in the Hierarchy, call it Game Manager and assign the player in the player slot on the script 🡪 this will help the enemy search and follow the player.

Alterations you can make.

In the scripts, you can alter some of the figures and behaviours: ***Enemy Controller Script***

* **Look Radius** – area the enemy will follow the player (Set at 20)
* **Attack Radius** – area the enemy will attack the player (Set at 10)
* **Forward Force** – the acceleration of the bullet going forward (Set at 35)
* **Up Force** – force of the bullet in height (Set at 20)
* **Time Between Attacks** – set the time between each bullet shot (set at 5)

Objects you must assign in the script:

* **Shoot Point** – an empty game object, set as a child to the enemy – decide where the enemy will shoot its bullets from
* **Projectile** – decide what the enemy shoots (a prefab of a bullet) – you must have a Rigidbody attached to projectile.

**Leader Board**

This documentation describes how to use the Leaderboard component in your project.

**Behaviours:**

* Table
* Leader Board
* Rank
* Score
* Name

**Table:**

This behaviour shows the data from each highscore. It should be placed in the boot scene as its used to start the game and only gets loaded once. It is made to appear on Awake. It has these properties:

* Number of entries in the table
* Minimum Score - lowest score randomly generated.
* Maximum Score – highest score randomly generated.
* Position – Rank (depending on if your score was the highest/lowest)
* Names – this initialises the names used in random score generation.

**Rank:**

This behaviour must be configured with three Unity UI Text objects. One each for the rank, name, and score fields. The function SetRank writes the given rank into the rank Text field as an integer. The function SetName writes the given name into the name Text field. The function SetScore writes the score as an integer into the score Text field, using the current default localisation settings for writing large numbers. For example, on a machine configured for the UK, the integer 1234567890 will display as 1,234,567,890.